Lucas Rezende Soares Cesar
Thor Franco Brenner

Implementação e integração de uma arquitetura de software para um crawler de postagens em plataformas digitais

Goiânia
2024

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): Lucas Rezende Soares Cesar e Thor Franco Brenner

Título do trabalho: Implementação e integração de uma arquitetura de software para um crawler de postagens em plataformas digitais

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Jacson Rodrigues Barbosa**, **Professor do Magistério Superior**, em 19/12/2024, às 16:59, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Lucas Rezende Soares Cesar** , **Discente**, em 19/12/2024, às 17:11, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Thor Franco Brenner**, **Discente**, em 19/12/2024, às 18:15, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5054721** e o código CRC **835E95EA**.

Referência: Processo nº 23070.046767/2024-32 SEI nº 5054721

Lucas Rezende Soares Cesar
Thor Franco Brenner

Implementação e integração de uma arquitetura de software para um crawler de postagens em plataformas digitais

Trabalho de conclusão de curso apresentado na Escola de Engenharia Elétrica, Mecânica e de Computação como requisito para a conclusão do curso de Engenharia de Computação e obtenção do título de bacharel em Engenharia de Computação

Orientador: Jacson Rodrigues Barbosa

Universidade Federal de Goiás - UFG

Escola de Engenharia Elétrica, Mecânica e de Computação (EMC)

Goiânia

2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Cesar, Lucas Rezende Soares
Implementação e integração de uma arquitetura de software para um crawler de postagens em plataformas digitais [manuscrito] / Lucas Rezende Soares Cesar, Thor Franco Brenner. - 2024.
XXV, 25 f.: il.

Orientador: Prof. Jacson Rodrigues Barbosa.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação (EMC), Engenharia da Computação, Goiânia, 2024.
Bibliografia. Apêndice.
Inclui gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Desinformação. 2. Crawlers. 3. Redes sociais. 4. Automação de dados. I. Brenner, Thor Franco . II. Barbosa, Jacson Rodrigues , orient. III. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Ao(s) dezenove dia(s) do mês de dezembro do ano de 2024 iniciou-se a sessão pública de defesa do Trabalho de Conclusão de Curso (TCC) intitulado "**Implementação e integração de uma arquitetura de software para um crawler de postagens em plataformas digitais**", de autoria de **Lucas Rezende Soares Cesar** e **Thor Franco Brenner**, do curso de Engenharia de Computação, do(a) EMC da UFG. Os trabalhos foram instalados pelo(a) Prof. Dr. Jacson Rodrigues Barbosa (INF/UFG) com a participação dos demais membros da Banca Examinadora: Prof. Dr. Marco Antonio Assfalk de Oliveira (EMC/UFG) e Prof. Dr. Valdemar Vicente Graciano Neto (INF/UFG). Após a apresentação, a banca examinadora realizou a arguição do(a) estudante. Posteriormente, de forma reservada, a Banca Examinadora atribuiu a nota final de 9,0, tendo sido o TCC considerado **Aprovado**.

Proclamados os resultados, os trabalhos foram encerrados e, para constar, lavrou-se a presente ata que segue assinada pelos Membros da Banca Examinadora.

Documento assinado eletronicamente por **Valdemar Vicente Graciano Neto** , **Professor do Magistério Superior**, em 19/12/2024, às 16:14, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Jacson Rodrigues Barbosa**, **Professor do Magistério Superior**, em 19/12/2024, às 16:14, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Marco Antonio Assfalk De Oliveira** , **Professor do Magistério Superior**, em 19/12/2024, às 20:03, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5054574** e o código CRC **C33FEEC5**.

Referência: Processo nº 23070.046767/2024-32 SEI nº 5054574

Implementação e integração de uma arquitetura de software para um crawler de postagens em plataformas digitais

Lucas R. S. Cesar
Escola de Engenharia Elétrica, Mecânica e de Computação (EMC)
Universidade Federal de Goiás
Goiânia - GO - Brasil
Email: lucascesar@discente.ufg.br

Thor F. Brenner
Escola de Engenharia Elétrica, Mecânica e de Computação (EMC)
Universidade Federal de Goiás
Goiânia - GO - Brasil
Email: thorbrenner@discente.ufg.br

Jacson R. Barbosa
Instituto de Informática (INF) - UFG
Universidade Federal de Goiás
Goiânia - GO - Brasil
Email: jacson_rodrigues@ufg.br

***Resumo*—O trabalho aborda a problemática da desinformação nas redes sociais, um fenômeno amplificado por tecnologias de inteligência artificial e estratégias de disseminação massiva, que comprometem a credibilidade de processos democráticos e a formação da opinião pública. A pesquisa propõe uma solução técnica para automatizar a coleta e análise de dados provenientes de agências de verificação de fatos e plataformas digitais, utilizando *crawlers* e APIs, com ênfase na integração com o Projeto Web 3.0.**

Os resultados obtidos demonstraram a eficácia da arquitetura desenvolvida em consolidar dados estruturados de fontes confiáveis e redes sociais, permitindo análises como a de sentimento para identificar polarizações e tendências sociais. Apesar das limitações impostas pelas APIs de redes sociais, o sistema mostrou-se escalável e funcional, contribuindo para o combate à desinformação de maneira mais ágil e acessível. Assim, o estudo ressalta a relevância da integração entre tecnologia e expertise humana para enfrentar desafios informacionais complexos.

Palavras-Chave: Desinformação, Crawlers, Redes sociais, Automação de dados

***Abstract*—The study addresses the issue of misinformation on social networks, a phenomenon exacerbated by artificial intelligence technologies and mass dissemination strategies, which undermine the credibility of democratic processes and the formation of public opinion. The research proposes a technical solution to automate the collection and analysis of data from fact-checking agencies and digital platforms, employing *crawlers* and APIs, with a focus on integration with the Web 3.0 Project.**

The results demonstrated the effectiveness of the developed architecture in consolidating structured data from reliable sources and social networks, enabling analyses such as sentiment analysis to identify polarizations and social trends. Despite limitations imposed by social network APIs, the system proved scalable and functional, contributing to a more agile and accessible fight against misinformation. Thus, the study highlights the importance of integrating technology and human expertise to tackle complex informational challenges.

Keywords: Misinformation, Crawlers, Social networks, Data automation

I. Introdução

As redes sociais começaram como uma forma de interação entre amigos e familiares, porém, com a evolução tecnológica, logo se expandiram para atender a diversos propósitos, hoje referem-se a uma variedade de tecnologias que facilitam o compartilhamento de textos, imagens, sons e vídeos [1]. De acordo com o Global Web Index, 46% dos usuários de internet em todo o mundo obtêm notícias por meio das redes sociais [2]. Isso se compara a 40% dos usuários que acessam notícias em sites de notícias.

No cenário eleitoral, as mídias sociais se tornaram indispensáveis para candidatos políticos e desempenham um papel estratégico, permitindo interação direta com seus eleitores, divulgação de suas propostas e mobilização apoiadores em larga escala. Além disso, as redes sociais digitais desempenham um papel crucial na formação da opinião pública, pois as informações compartilhadas nessas plataformas têm o poder de direcionar debates, influenciar percepções e moldar comportamentos coletivos.

Ao atuar como um espaço dinâmico para o diálogo público, elas se tornam ferramentas indispensáveis para a construção e transformação de narrativas sociais. Dessa forma, a influência das mídias sociais transcende o âmbito individual, impactando profundamente a sociedade e os negócios, e consolidando-se como uma força global transformadora.

A influência das redes sociais é grande, mas, por outro lado, elas também trazem desafios significativos, como a disseminação de notícias falsas. A combinação da facilidade de compartilhar informações com o enorme alcance dessas plataformas torna as redes sociais um terreno fértil para a propagação de desinformação.

Esse fenômeno é particularmente preocupante em contextos como eleições, onde as notícias falsas podem manipular a opinião pública e comprometer a credibilidade dos processos democráticos. Notícias falsas frequentemente exploram o viés de confirmação dos usuários, levando-os a compartilhar informações que estão alinhadas com suas crenças, sem verificar a sua veracidade.

As notícias falsas sempre estiveram presentes na história, embora sua denominação, os meios de divulgação e seu potencial persuasivo tenham-se transformado significativamente nos últimos anos. Desde os primórdios, é discutível que os

Cavaleiros Templários foram derrubados devido às notícias falsas da época [3]. Em 2016, o uso do termo *fake news* ficou mundialmente conhecido na época das eleições dos Estados Unidos da América, época em que os eleitores de Donald Trump produziram e compartilharam intensamente conteúdos falsos sobre a candidata Hillary Clinton. No ano de 2017, *fake news* foi nomeada a palavra do ano pelo dicionário inglês britânico Collins [4].

No cenário brasileiro, o fenômeno das *fake news* é entendido como qualquer produção e disseminação de notícias sabidamente falsas por meio de veículos de comunicação, redes sociais ou aplicativos de mensagens, com o objetivo de desinformar e atrair a atenção do público para obter vantagem econômica, política ou ideológica. De acordo com a Academia Brasileira de Letras [5], o conceito de pós-verdade está intimamente relacionado a essa prática, uma vez que as *fake news* apelam para as emoções, reforçando crenças preexistentes, em detrimento da verdade. Nesse contexto, a opinião pública demonstra menos preocupação com a veracidade dos fatos e se concentra mais naquilo que deseja acreditar, reforçando preferências individuais em detrimento da objetividade.

A eficácia das *fake news* está diretamente ligada à rapidez e ao alcance proporcionados pelas plataformas digitais. Por meio de algoritmos que favorecem o engajamento, conteúdos falsos frequentemente alcançam maior visibilidade do que informações verificadas. Além disso, a repetição constante dessas narrativas contribui para a sua assimilação como verdade por parte de determinados grupos. Esse mecanismo é amplificado por bolhas informacionais, onde os indivíduos consomem apenas conteúdos alinhados às suas opiniões, reduzindo a possibilidade de confronto com fontes confiáveis e diversas. Assim, as *fake news* não apenas desinformam, mas também fragmentam o debate público, minando a confiança nas instituições e no jornalismo tradicional.

O combate às informações falsas é essencial em uma sociedade cada vez mais impactada pela desinformação. Agências como G1, UOL e Lupa monitoram declarações públicas, postagens em redes sociais e mensagens em aplicativos, verificando a veracidade de informações e classificando-as como falsas, verdadeiras ou "meias-verdades".

Esse trabalho, feito manualmente, é eficiente, mas demorado e repetitivo, o que torna-se um problema diante do enorme volume de informações geradas diariamente. Para enfrentar esse desafio, este estudo propõe automatizar o processo com *web crawlers* e APIs, consolidando as verificações em um único sistema. Assim, seria possível combater a desinformação de forma mais rápida e acessível, ampliando o alcance dos resultados para a sociedade.

A organização deste trabalho está estruturada em seções que detalham todas as etapas realizadas. A introdução apresenta o problema e os objetivos do estudo. A seção II aborda os conceitos relacionados à verificação de *fake news* e tecnologias digitais. Na seção III, são descritos os métodos de pesquisa utilizados. As seções IV e V tratam, respectivamente, da pesquisa inicial e do desenvolvimento do *web crawler*, enquanto a seção VI apresenta a arquitetura de software proposta. Na seção VII, é realizada a avaliação da arquitetura desenvolvida. A seção VIII apresenta os resultados e discussões, seguida pela seção IX, que aborda as limitações do estudo. Finalmente, a seção X traz as conclusões finais, destacando as contribuições do trabalho e possibilidades de estudos futuros. Além disso, os apêndices fornecem detalhes técnicos, como códigos utilizados, exemplos de raspagem de dados e tabelas complementares, oferecendo suporte para a replicabilidade e o aprofundamento das soluções desenvolvidas.

A. Motivação: Projeto Web 3.0

A principal motivação deste estudo foi a integração com o Projeto Web 3.0, também denominado DAurora, uma iniciativa fruto de um convênio entre a Agência Nacional de Telecomunicações (ANATEL) e a Universidade Federal de Goiás (UFG). O projeto visa desenvolver uma Prova de Conceito (PoC) descentralizada para fornecer instrumentos de identificação, classificação e contenção de *fake news* e ações que levam à disseminação da desinformação, propiciando uma avaliação do potencial tecnológico para dar respostas efetivas para o problema [6].

A Figura 1 ilustra a arquitetura geral do Projeto Web 3.0 [7], destacando em vermelho a área onde este estudo se insere. O papel principal desta pesquisa no projeto foi desenvolver um sistema de *Web Crawling* capaz de realizar buscas automatizadas em agências de notícias e plataformas de *fact-checking*, complementado pela integração com APIs para coletar dados relevantes provenientes de Redes Sociais Digitais. A escolha pelo desenvolvimento do *Web Crawling* e pela integração de APIs foi motivada pela necessidade de criar um pipeline escalável e confiável, que não apenas auxiliasse na validação de informações, mas também alimentasse o ecossistema do DAurora com dados estruturados e categorizados.

B. Objetivos

As seções a seguir apresentam os objetivos do estudo:

1) Objetivo Geral: Construir de um sistema de coleta e análise de dados, composto por um *crawler* para verificar informações provenientes de agências de *fact-checking* e redes sociais, aliado à análise das postagens coletadas.

2) Objetivos Específicos: Desenvolver um sistema de *Web Crawling* capaz de coletar dados de agências de *fact-checking*, estruturando as informações de forma eficiente para que estejam prontamente disponíveis para consulta e utilização no ecossistema do Projeto Web 3.0.

Desenvolver um módulo de coleta de dados por meio de requisições a APIs da rede social YouTube, permitindo a extração de informações relevantes.

Desenvolver uma análise de sentimento aplicada aos dados coletados, com categorização para enriquecer as análises e subsidiar tomadas de decisão no âmbito do projeto.

II. Conceitos relacionados à verificação de *fake news* nas plataformas digitais

Redes Sociais Digitais - São divididas em algumas categorias — redes sociais, notícias sociais, compartilhamento de

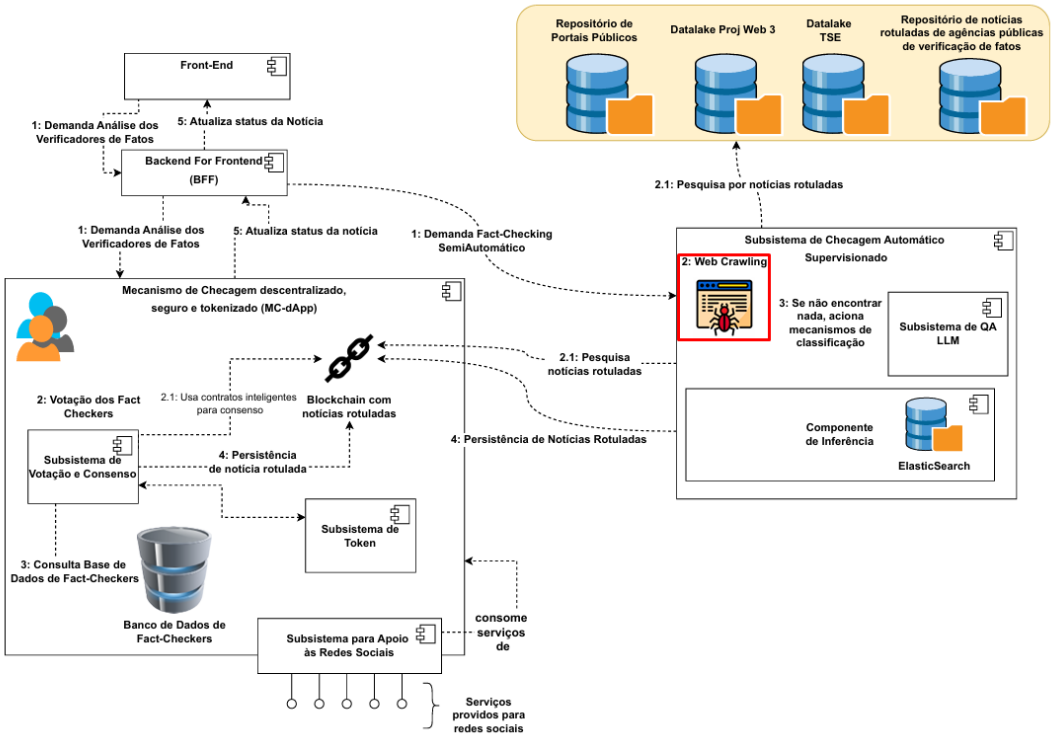

Fig. 1: Arquitetura geral do DAurora, adaptada de [7]

mídia, blogs e fóruns online, que desempenham um papel essencial tanto na vida pessoal quanto profissional. Essas plataformas atendem a diversos interesses, desde conectar indivíduos com opiniões similares ao redor do mundo até servir como ferramentas de engajamento para artistas, políticos e corporações.

Web Crawler - Também conhecido como robô, *spider* ou *worm*, é um programa ou *script* automatizado que navega pela *World Wide Web* de forma sistemática. A estrutura da WWW é representada como um gráfico de linha, onde as páginas web são os nós e os hiperlinks são as linhas que ligam esses pontos. Isso permite que o processo de busca seja resumido a uma travessia por um grafo direcionado. Ao seguir os links de uma página, um *web crawler* pode acessar e explorar novas páginas a partir de uma página inicial. Os *scripts* se movem de página em página usando essa estrutura interligada e armazenam as páginas visitadas em um repositório local. A função principal de um *crawler* é criar uma réplica das páginas acessadas para que um mecanismo de busca possa processá-las, indexá-las e, assim, facilitar a busca rápida por informações. Mecanismos de busca armazenam informações de várias páginas da internet, as quais foram coletadas pelo *web crawler*, um navegador automatizado que segue os links que encontra.

Python - É uma linguagem de programação interpretada, interativa e orientada a objetos [8]. *Python* oferece aos desenvolvedores uma combinação de recursos avançados e uma sintaxe acessível e fácil de entender. Ele dispõe de interfaces para diversas chamadas e bibliotecas do sistema, além de suporte a diferentes sistemas de janelas. A linguagem é altamente extensível, permitindo integração com linguagens como C e C++.

Framework - É uma estrutura que serve como base no desenvolvimento de software. Isso permite que você não precise começar completamente do zero, tornando o processo de desenvolvimento mais eficiente. Geralmente, um *framework* está associado a uma linguagem de programação específica e é projetado para tarefas específicas. No desenvolvimento de software, um *framework* oferece uma base confiável, pois foi projetado e testado por outros desenvolvedores. Isso ajuda a reduzir o tempo de desenvolvimento e diminui o risco de erros. Como não é necessário escrever tudo do zero, as chances de cometer erros diminuem. Além disso, por já ter sido testado, ele proporciona maior segurança e confiança no seu uso.

Scrapy - É um *framework* rápido e de alto nível para *web crawling* e *web scraping*, usado para rastrear sites e extrair dados estruturados de suas páginas. Pode ser utilizado para uma ampla gama de finalidades, desde mineração de dados até monitoramento e testes automatizados [9].

Script - É um conjunto de instruções ou comandos escritos em uma linguagem de programação, projetado para ser interpretado e executado por um programa específico ou ambiente de execução. Diferentemente de programas compilados, os *scripts* geralmente não precisam de um processo de compilação antes de serem executados, sendo interpretados diretamente.

Airflow - É uma plataforma de orquestração de fluxos

de trabalho de dados, possibilita o agendamento, monitoramento e execução de tarefas de forma organizada e escalável, também proporciona mecanismos de monitoramento robustos, facilitando a detecção de falhas e a reexecução automática das tarefas quando necessário, garantindo a confiabilidade do processo.

VADER/BERTopic - No contexto de análise de sentimento, VADER (*Valence Aware Dictionary and Sentiment Reasoner*) e BERTopic (*Bidirectional Encoder Representations for Topic Modeling*) são duas ferramentas usadas para essas análises. O VADER é baseado em regras e léxicos, ou seja, foi especialmente projetado para detectar sentimentos em texto [10]. Ele utiliza um dicionário pré-construído de palavras associadas a pontuações de sentimento e consegue reconhecer as nuances do texto, como palavras em maiúsculas, emojis, exclamações e palavras de intensificação (e.g. "muito bom" possui maior impacto do que apenas "bom"). No caso do BERTopic, é utilizado o aprendizado de máquina não supervisionado para identificar temas ou tópicos latentes em conjuntos de texto [11]. Aplica modelos para transformar frases ou textos em vetores numéricos que capturam o significado semântico, depois aplica algoritmos de agrupamento e assim extrai as palavras mais relevantes para cada sentimento.

Banco de Dados - É uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados [12].

API - A sigla significa *Application Programming Interface* (Interface de Programação de Aplicações), é um conceito que se refere a um conjunto de regras e padrões que permitem a comunicação entre diferentes softwares. No contexto de APIs, o termo "aplicação" se refere a qualquer sistema ou programa com uma funcionalidade específica [13]. A interface funciona como um acordo entre os sistemas envolvidos, especificando como devem interagir por meio de solicitações e respostas. A documentação das APIs fornece orientações detalhadas para os desenvolvedores sobre a forma correta de construir essas solicitações e interpretar as respostas. APIs são ferramentas que permitem a comunicação entre diferentes componentes de software, estabelecendo um conjunto de definições e protocolos para essa interação. Elas atuam como pontes que conectam sistemas distintos, possibilitando a troca de informações. Assim, a API funciona como um meio estruturado de comunicação, com regras definidas que garantem que os sistemas envolvidos possam solicitar e fornecer informações de maneira eficaz.

Inteligência Artificial - refere-se à capacidade de máquinas e sistemas computacionais de imitar a inteligência humana, realizando tarefas como percepção, reconhecimento, aprendizado, resposta e resolução de problemas. A IA é uma tecnologia usada para analisar dados, identificar padrões e auxiliar na tomada de decisões. No campo educacional, tem sido aplicada para oferecer experiências de aprendizado personalizadas e monitorar o progresso dos estudantes. A IA se posiciona como uma ferramenta poderosa para melhorar a qualidade da educação [14].

Arquitetura de Software - refere-se à estrutura fundamental de um sistema de software, incluindo a organização de seus componentes, suas interações e os princípios que orientam seu design e evolução. Este é um aspecto crucial no desenvolvimento de software, pois define como os diferentes elementos do sistema se combinam e se comunicam, influenciando diretamente o desempenho, a confiabilidade, os custos e a adaptabilidade [15].

LLM (*Large Language Models*) - Os LLMs (Modelos de Linguagem de Grande Escala) são ferramentas de inteligência artificial (IA) projetadas para gerar textos semelhantes aos produzidos por humanos, baseados em vastas quantidades de dados. Esses modelos utilizam redes neurais recorrentes em camadas múltiplas, revolucionando a capacidade de compreender e gerar texto. Diferentemente dos modelos linguísticos tradicionais, que utilizavam métodos estatísticos para prever a próxima palavra, os LLMs empregam modelos baseados em transformadores, permitindo o processamento paralelo de grandes volumes de dados. Como resultado, esses modelos conseguem produzir textos mais naturais e fluentes [16].

Machine Learning - É o processo de criar algoritmos que aprendem padrões com base em dados, permitindo previsões e tomadas de decisão. Este livro explica conceitos fundamentais, como ajuste de curvas, teoria das probabilidades, modelos estatísticos, redes neurais, aprendizado supervisionado e aprendizado não supervisionado [17].

III. Método de Pesquisa

O processo iniciou com uma pesquisa exploratória em artigos científicos, notícias e bibliotecas especializadas, buscando embasamento teórico e técnico. A partir dessa pesquisa bibliográfica, foram desenvolvidas soluções técnicas que direcionaram a construção da arquitetura de software. Essa arquitetura foi projetada para garantir a integração eficiente entre o sistema de coleta e a aplicação DAurora.

Para o desenvolvimento do *crawler*, foi necessário realizar um levantamento criterioso das agências de *fact-checking* a serem utilizadas na coleta de dados. O principal critério para a seleção foi a confiabilidade das agências, visando garantir que os dados coletados sejam consistentes e seguros. A escolha das agências foi baseada na análise de suas metodologias e reputação no combate à desinformação.

Após a seleção das agências, iniciou-se o processo de definição das ferramentas técnicas para implementação do *crawler*, incluindo a escolha da linguagem de programação e do *framework* mais adequados para a extração de dados. Esses elementos foram selecionados com base em critérios como desempenho, compatibilidade com as fontes de dados e facilidade de integração com a arquitetura do Projeto Web 3.0.

Paralelamente, no âmbito das Redes Sociais Digitais, o processo envolveu a análise de quais plataformas seriam uti-

lizadas para coleta de dados, levando em consideração fatores como relevância no contexto de disseminação de informações e disponibilidade de APIs que permitam acesso estruturado aos dados. Além disso, foi avaliada a viabilidade técnica de integração dessas APIs ao sistema, de forma a garantir que os dados sejam extraídos e organizados de maneira eficiente.

Após a definição das ferramentas de desenvolvimento, foi realizada uma análise detalhada sobre quais dados seriam coletados tanto das agências de *fact-checking* quanto das Redes Sociais Digitais. Dado o imenso volume de informações disponíveis, tornou-se essencial um estudo aprofundado para identificar quais dados são mais relevantes e pertinentes ao objetivo do projeto. Essa etapa incluiu a priorização de informações diretamente relacionadas à desinformação e ao contexto político e eleitoral.

Ademais, foi definida a forma de organização e armazenamento dos dados coletados. Decidiu-se por uma estrutura que permita tanto o fácil acesso quanto a integridade das informações, garantindo que os dados estejam prontos para futuras análises. Nesse processo, foram definidos os formatos de saída das informações, o local de armazenamento (como diretórios ou bancos de dados) e as estratégias para garantir a consistência e acessibilidade dos dados ao longo do tempo.

Com os dados devidamente organizados, a próxima etapa envolveu o planejamento e aplicação de técnicas de análise de sentimento. Inicialmente, foi realizado um estudo das ferramentas disponíveis, avaliando suas capacidades de processamento linguístico, suporte a múltiplos idiomas e adaptabilidade ao contexto dos dados coletados. A escolha da ferramenta considerou fatores como precisão, escalabilidade e facilidade de integração com o sistema já desenvolvido.

Um ponto essencial desse processo foi identificar onde a análise de sentimento seria mais relevante. Após uma avaliação criteriosa, concluiu-se que, as notícias das agências de *fact-checking* já são rotuladas por especialistas, a análise de sentimento teria maior valor agregado em comentários e postagens de Redes Sociais Digitais. Nesses casos, as emoções e opiniões expressas pelos usuários poderiam fornecer *insights* valiosos sobre o impacto e a disseminação de desinformação, além de identificar polarizações e tendências no comportamento social.

Este artigo aborda as principais etapas do estudo, englobando a pesquisa, o desenvolvimento, a análise de resultados, a integração com o Projeto Web 3.0 e as conclusões finais. Durante todo o processo, foi empregada uma abordagem iterativa e incremental, permitindo ajustes contínuos para atender aos requisitos do projeto e superar desafios técnicos, assegurando a consistência e a relevância dos resultados obtidos.

IV. Pesquisa Inicial

Durante as leituras iniciais, foi compreendido que as agências de verificação de fatos desempenham um papel fundamental no combate à desinformação. Buscou-se entender como esses portais especializados verificam a veracidade de postagens, notícias, vídeos e outros conteúdos digitais. Essa análise evidenciou que o trabalho dessas agências é amplamente manual, baseado em processos rigorosos de validação que incluem o uso de dados oficiais, entrevistas com especialistas e análises contextuais. Os conteúdos são então classificados com rótulos que, embora subjetivos, asseguram informações confiáveis aos leitores. Além disso, observou-se que o impacto das verificações é significativo na sociedade, contribuindo para mitigar os danos causados pela disseminação de informações falsas [18].

Paralelamente, foi realizada uma pesquisa técnica para orientar o desenvolvimento da solução prática. Essa etapa inicial incluiu o estudo detalhado do funcionamento de um *Web Crawler*, compreendendo suas capacidades e limitações no contexto da coleta automatizada de dados. Além disso, foram definidas a linguagem de programação e o *framework* que suportariam a implementação, garantindo que as ferramentas selecionadas estivessem alinhadas aos objetivos e requisitos do projeto.

A linguagem de programação utilizada para o desenvolvimento do sistema, optou-se por *Python* devido à sua simplicidade e versatilidade, características que tornam a linguagem ideal para a criação de *scripts*, que é o foco deste estudo. Além disso, *Python* oferece uma ampla gama de *frameworks* e bibliotecas consolidadas para tarefas de coleta e análise de dados, o que facilita a implementação de soluções robustas e eficientes.

Após a definição da linguagem de programação, iniciou-se a busca pelo *framework* mais adequado, considerando os critérios de popularidade, facilidade de integração e compatibilidade com o projeto. Três *frameworks* foram inicialmente analisados: *BeautifulSoup*, *Scrapy* e *Kimura*. Embora o *Kimura* apresentasse características interessantes, foi descartado por ser desenvolvido em *Ruby*, o que não se alinhava à escolha do *Python* como linguagem base. A seleção ficou então entre o *BeautifulSoup* e o *Scrapy*, que apresentam funcionalidades semelhantes. No entanto, optou-se pelo *Scrapy* devido à sua maior popularidade, ampla base de usuários e volume de estudos disponíveis, facilitando a continuidade da pesquisa.

O fluxo de funcionamento de um *web crawler* com *Scrapy* segue os passos descritos [19], conforme indicado na Figura 2. O processo começa no *Spider* (1), que gera as requisições iniciais com as URLs a serem rastreadas. Essas requisições são enviadas para a *Engine*, que as repassa para o *Scheduler* (2). O *Scheduler* organiza essas requisições em uma fila para processamento futuro e, quando solicitado pela *Engine*, devolve as próximas requisições a serem processadas (3).

A *Engine* então envia essas requisições ao *Downloader*, mas antes elas passam pelos *Downloader Middlewares*, onde podem ser alteradas, por exemplo, para incluir cabeçalhos ou autenticações. O *Downloader* (4) busca as páginas correspondentes às URLs, obtendo as respostas com o conteúdo das páginas, e as devolve para a *Engine*, novamente passando pelos *Downloader Middlewares* para eventuais ajustes ou manipulações (5).

Após a *Engine* receber as respostas do *Downloader*, elas são enviadas ao *Spider*, passando pelos *Spider Middlewares*, que podem realizar modificações ou filtragens nas respostas antes de entregá-las ao *Spider* (6). No *Spider* (7), as respostas

são analisadas de acordo com a lógica definida pelo usuário. Ele extrai dados (itens) das páginas e, caso necessário, gera novas requisições para explorar links relacionados.

Esses itens extraídos são enviados pela *Engine* para o *Item Pipeline* (8), onde passam por processos como validação, limpeza e persistência, sendo salvos em bases de dados ou arquivos. As novas requisições geradas pelo *Spider* retornam ao *Scheduler*, reiniciando o ciclo até que todas as requisições sejam processadas. Esse fluxo contínuo garante que o *crawler* percorra todas as páginas necessárias e colete os dados desejados de maneira eficiente e estruturada.

Com a parte do *Web Crawler* definida, o estudo avançou para outro ponto crucial: o uso de APIs em Redes Sociais Digitais. Foi realizada uma análise das principais plataformas, incluindo Facebook, Instagram, Twitter/X, YouTube e TikTok, para determinar qual seria a mais adequada para o estudo. Facebook e Instagram, ambos sob a gestão da Meta, assim como o TikTok, foram descartados devido à necessidade de um processo manual de verificação e aprovação diretamente pelas empresas [20, 21], o que poderia atrasar significativamente o andamento do projeto. O Twitter/X, por sua vez, estava temporariamente banido no Brasil durante o período de realização deste estudo [22], tornando sua utilização inviável. Diante dessas restrições, a escolha recaiu sobre o YouTube, que oferece uma API fornecida pelo Google [23]. Apesar de possuir limitações de uso relacionadas ao orçamento, a API permite acesso prolongado e funcional com um e-mail pessoal, atendendo às necessidades do projeto de forma prática e eficiente.

V. Desenvolvimento do Web Crawler

A primeira etapa prática consistiu na criação de *crawlers* capazes de coletar dados de portais especializados em *fact-checking*. Para isso, utilizou-se a biblioteca *Scrapy* da linguagem *Python*. Os portais escolhidos — G1 Fato ou Fake, Agência Aos Fatos, Agência Lupa, UOL Confere e Boatos.org — foram selecionados com base na credibilidade e no impacto de suas verificações.

O primeiro portal utilizado para o desenvolvimento do *spider* foi a Agência Aos Fatos. Inicialmente, foi analisada a estrutura HTML do site para identificar os elementos e padrões necessários à extração dos dados. Em um primeiro teste, foram coletados apenas dois elementos principais: o título das matérias e a classificação de sua veracidade (se era falso ou verdadeiro), conforme destacado na Figura 3. Essa abordagem inicial teve como objetivo validar o funcionamento do *spider* e garantir que a extração dos dados fosse realizada de forma consistente, preparando o sistema para coletas mais completas em etapas posteriores.

Com a ferramenta funcionando e os primeiros dados coletados com sucesso, o próximo passo foi aprimorar o *spider* para acessar cada notícia individualmente. Nessa etapa, o foco foi expandir o escopo da coleta de dados, extraindo informações mais detalhadas, como a manchete, o autor, a data de publicação, a classificação da notícia (*tag* de verdadeiro ou falso) e o conteúdo completo da matéria. Esse avanço permitiu capturar um conjunto mais robusto de dados, essencial para análises posteriores e para atender aos objetivos do estudo.

No caso deste estudo, a saída dos dados foi configurada para ser gerada em arquivos no formato .JSON, onde cada arquivo representava os dados de uma fonte específica. Essa estrutura permitiu a separação lógica dos dados coletados, facilitando tanto o armazenamento quanto o processamento posterior, garantindo a integridade e a organização necessárias para o projeto.

Após o desenvolvimento do *spider* para a Agência Aos Fatos estar praticamente concluído, código apresentado no Apêndice B, Código 1, o estudo avançou para as demais agências de *fact-checking*. Cada uma apresentou desafios específicos devido às diferenças nas tecnologias utilizadas em seus sites. Foi necessário criar *spiders* personalizados para cada portal, pois aspectos como estruturas HTML dinâmicas e carregamento assíncrono de conteúdo tornaram a extração dos dados mais complexa.

Um dos principais desafios enfrentados foram sites que utilizam *JavaScript* para renderizar informações essenciais, o que impossibilitou a coleta direta com ferramentas tradicionais. Para superar essa limitação, foi integrada ao projeto a biblioteca *Playwright*, desenvolvida para atender necessidades de testes de ponta a ponta. Essa ferramenta oferece suporte a mecanismos modernos de renderização, como *Chromium*, *WebKit* e *Firefox*, permitindo a renderização completa das páginas antes da coleta [25]. Além disso, o *Playwright* possibilita testes em múltiplos sistemas operacionais (Windows, Linux e macOS), tanto localmente quanto em pipelines de integração contínua (CI), e pode ser executado em modo *headless* ou com interface gráfica, além de suportar a emulação nativa de dispositivos móveis [26].

Outro desafio encontrado foi a diversidade de rótulos utilizados pelas agências de *fact-checking* para classificar as notícias verificadas, com termos como “falso”, “não é bem assim”, “impreciso” ou “verdadeiro”. Essa diversidade tornou a categorização dos dados um ponto central do estudo, exigindo uma análise detalhada das taxonomias empregadas por cada agência para justificar seus vereditos. O objetivo foi criar uma estrutura padronizada que permitisse integrar informações de diferentes fontes sem perder a consistência semântica, garantindo que os dados representassem a realidade de forma precisa.

Os dados extraídos foram mantidos na forma original, preservando as classificações específicas de cada agência. A interpretação e possível simplificação desses rótulos para categorias mais gerais, como “Verdadeiro” ou “Falso”, foram deixadas como uma responsabilidade futura a ser realizada pelo ecossistema do Projeto Web 3.0, que utilizará esses dados para análises mais avançadas. Essa decisão garantiu que a coleta não comprometesse a riqueza de informações fornecidas por cada fonte.

A. Escalabilidade e Automação com Apache Airflow

A escalabilidade dos *web crawlers* demanda ferramentas de automação sofisticadas, sendo o *Apache Airflow* uma

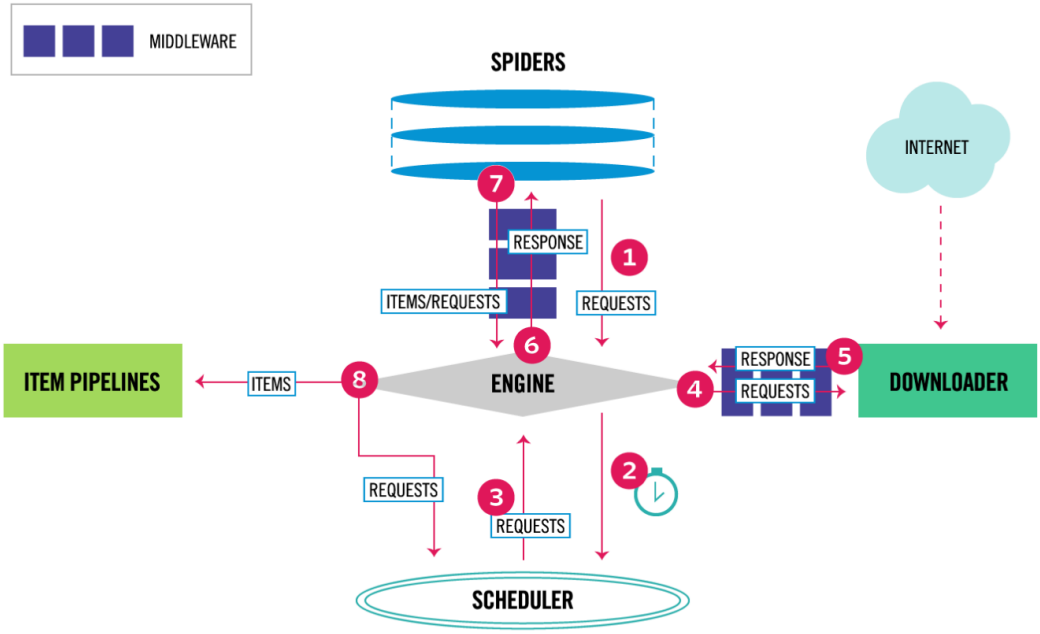

Fig. 2: Fonte: Documentação Biblioteca *Scrapy* [19]

Fig. 3: Fonte: Site Oficial do Aos Fatos [24]

solução tecnológica para expandir a eficiência e o alcance dos processos de coleta e indexação de dados na web. Esta plataforma emerge como uma estratégia crítica para gerenciar a complexidade inerente à coleta automatizada de informações em larga escala.

No contexto específico deste trabalho, a construção de uma base de dados consistente e permanentemente atualizada constitui um elemento basilar para analisar e mitigar os impactos da desinformação em períodos de alta volatilidade informacional, como ciclos eleitorais. A desinformação caracteriza-se por sua natureza altamente dinâmica e capacidade de disseminação exponencial através de plataformas digitais e redes sociais. A geração contínua de novos conteúdos, frequentemente adaptados a eventos contemporâneos, demandas sociais emergentes ou tendências políticas em curso, exige abordagens metodológicas robustas de monitoramento e análise.

A implementação do *Apache Airflow* como ferramenta de gerenciamento de execução automatizada dos *crawlers* representou uma solução arquitetural estratégica. Mediante a organização do fluxo de trabalho em DAGs (*Directed Acyclic Graphs*), foi possível estruturar a execução agendada dos *spiders*, garantindo a atualização contínua da base de dados sem intervenção manual. Essa abordagem mostrou-se essencial para processar o volume crescente de informações e atender à demanda por dados precisos e tempestivos sobre fenômenos de desinformação.

A implementação da DAG no *Airflow* envolve a definição de tarefas organizadas para gerenciar a execução de *spiders Scrapy* e *scripts* relacionados à coleta de dados no Youtube. Define-se a DAG propriamente dita, especificando uma descrição, o intervalo de execução programada e o comportamento de recuperação de execuções passadas. A primeira tarefa, utilizando o operador *Bash*, clona o repositório contendo os *spiders* em um diretório temporário. Após essa etapa, cada *spider* é executado em sequência por meio de comandos *Bash* que navegam até os diretórios específicos e iniciam as execuções dos *spiders*, salvando os resultados

em arquivos JSON. Tarefas adicionais incluem a execução de *scripts* que utilizam APIs como a YouTube Data API v3 e outras relacionadas. Todas as tarefas são encadeadas de forma lógica: a clonagem do repositório é a etapa inicial, seguida pela execução simultânea dos *spiders* e, finalmente, pelas tarefas relacionadas às APIs. A estrutura modular e organizada da DAG garante que cada passo dependa do anterior, mantendo o fluxo de trabalho robusto e automatizado.

A arquitetura de execução implementada no *Airflow* foi projetada para garantir a atualização contínua e a organização dos resultados. Sempre que um *script* é acionado, o processo inicial inclui o clone do repositório que contém o código mais recente, assegurando que a versão mais atualizada esteja sendo utilizada. Em seguida, cada *script* é executado individualmente, processando suas respectivas tarefas de coleta e processamento de dados. Os resultados obtidos são armazenados localmente no formato .JSON, o que facilita a organização e o acesso posterior para análises.

B. Expansão para Redes Sociais Digitais

Reconhecendo o papel das redes sociais na disseminação de desinformação, o escopo do trabalho foi expandido para incluir a coleta de dados do YouTube, uma plataforma de grande alcance e alta relevância no cenário digital. Para isso, foi utilizada a YouTube Data API v3, que possibilitou a busca de vídeos com base em *queries* específicas, permitindo ordenar os resultados por relevância e limitar a quantidade de itens extraídos, otimizando o uso dos recursos disponíveis.

No contexto deste estudo, a *query* utilizada foi "Eleições Goiânia 2024", selecionada de forma fixa e específica para fins de teste. Essa escolha permitiu avaliar o funcionamento do sistema de coleta de dados em um cenário controlado, garantindo a consistência e a confiabilidade durante o desenvolvimento e a validação inicial. No entanto, ao integrar o sistema com o Projeto Web 3.0, o parâmetro de busca será configurado como dinâmico, permitindo maior flexibilidade na definição das *queries*, de acordo com as necessidades específicas de análise e pesquisa do projeto.

Os metadados coletados incluíram informações valiosas, como o ID do vídeo, título, descrição, data de publicação, duração, número de curtidas, favoritos e métricas relacionadas aos comentários, o que proporcionou uma visão abrangente do conteúdo.

Visando aprofundar a análise para além dos metadados superficiais, implementou-se um processo de transcrição automatizada utilizando a *YouTube Transcript API*. Essa ferramenta possibilitou a geração de transcrições textuais completas a partir dos identificadores dos vídeos, agregando uma camada adicional de profundidade analítica ao processo de coleta.

Outrossim, foi testada a possibilidade de gravar os vídeos coletados para futuras análises no Projeto Web 3.0, especialmente no contexto de estudos relacionados a *deep fakes*. Para esse propósito, foi utilizada a yt_dlp API, uma ferramenta eficiente para o download de vídeos. No entanto, embora a ideia tenha sido implementada aqui neste estudo, ela não foi continuada no Projeto Web 3.0, pois sua aplicação iria contra as diretrizes de uso do YouTube, bem como as regulamentações de direitos autorais, o que poderia comprometer a conformidade ética e legal do projeto.

A implementação dessa metodologia enfrentou desafios técnicos significativos, particularmente no que concerne às restrições impostas pela API. Os limites de requisição, políticas de segurança e protocolos de acesso demandaram um planejamento metodológico refinado para a otimização das consultas e garantia da eficiência do processo de coleta. Além disso, a integração do sistema com o *Apache Airflow* foi estendida para incluir os *scripts* de extração e transcrição, mantendo o fluxo de trabalho completamente automatizado e centralizado.

C. Integração Final e Organização dos Dados

A etapa final de desenvolvimento do *crawler* caracterizou-se pela consolidação e integração dos dados coletados de múltiplas fontes, configurando uma arquitetura metodológica, capaz de obter metadados de no mínimo, um ano de notícias de cada agência de *fact-checking* e 500 vídeos do Youtube, por execução da DAG no *Airflow*. A decisão de padronizar o armazenamento em formato JSON representou uma estratégia técnica fundamental para garantir a interoperabilidade e a consistência dos dados coletados.

A escolha do formato JSON como padrão de armazenamento foi baseada em sua adequação ao contexto de treinamento de modelos de inteligência artificial. Além disso, o JSON possui uma estrutura flexível que permite representar dados complexos e aninhados, como metadados heterogêneos de vídeos (títulos, descrições, comentários), transcrições completas e resultados de verificações de fatos. Antes da decisão final, foram analisados outros formatos, como XML e CSV. O XML, embora também suporte hierarquia, foi descartado devido à sua maior complexidade e tamanho. Já o CSV, apesar de eficiente para dados tabulares, não oferece suporte nativo para representações aninhadas, tornando-o menos ideal para armazenar dados ricos e variados necessários nesse contexto.

VI. A ARQUITETURA DE SOFTWARE PROPOSTA

A arquitetura de software proposta para este estudo está detalhada nas Figuras 4 e 5. A visão de sistema apresenta os sistemas que compõem o projeto, destacando os serviços oferecidos e as estruturas de armazenamento. Essa visão ilustra como cada sistema opera de maneira integrada, com funções especializadas, como coleta de dados, processamento, análise de sentimento e monitoramento, além de evidenciar os bancos de dados onde as informações coletadas são armazenadas.

Por outro lado, a visão lógica descreve a lógica do projeto de forma organizada e modular, dividida em duas camadas principais: a Camada de Aplicação e a Camada de Persistência. A Camada de Aplicação abrange os processos de coleta de dados, processamento e análise, enquanto a Camada de Persistência concentra-se na execução dos fluxos de trabalho (gerenciados pelo *Airflow*) e no armazenamento final dos resultados em arquivos JSON. Essa divisão em camadas reflete uma arquitetura bem estruturada, que permite organização,

escalabilidade e clareza no fluxo de dados e operações do sistema.

A visão dos sistemas apresentada na Figura 4 destaca os sistemas presentes. O Sistema Coletor de Dados (*Web Crawler*) é o responsável por extrair informações tanto de agências verificadoras de fatos quanto de redes sociais, como o YouTube. Ele trabalha de forma integrada com o Sistema Gerenciador de Execução, implementado pelo *Apache Airflow*, que orquestra todas as tarefas de coleta e monitoramento. O Sistema de Processamento de Dados entra em ação após a coleta, realizando a limpeza, transformação e organização dos dados para torná-los consistentes e prontos para análise. Paralelamente, o Sistema de Monitoramento de Métricas acompanha o desempenho dos fluxos, garantindo eficiência e apontando possíveis gargalos no processo.

A análise dos dados é conduzida pelo Sistema de Análise de Sentimento, que utiliza as bibliotecas VADER e BERTopic para identificar padrões e classificar as informações, complementando o trabalho das etapas anteriores. Por fim, o Sistema de Armazenamento de Metadados assegura que os dados estruturados estejam devidamente organizados e acessíveis para análises futuras.

Os serviços do sistema incluem a coleta de notícias rotuladas, que extrai dados de agências verificadoras. O serviço de coleta de dados do vídeo realiza a extração de metadados dos vídeos, incluindo título, descrição e comentários, fornecendo uma visão abrangente do conteúdo disponível na plataforma. Além disso, o serviço de transcrição de vídeo processa os vídeos coletados, gerando transcrições completas do conteúdo audiovisual. Por fim, o serviço de análise de sentimento dos dados dos vídeos coletados aplica as funções das bibliotecas já citadas para interpretar e categorizar o sentimento presente, tanto nas transcrições dos vídeos quanto nos comentários associados.

O armazenamento é realizado em duas bases principais: o DB Agências Verificadoras, que mantém os dados coletados de fontes como Aos Fatos e Agência Lupa, e o DB YouTube, que concentra os metadados e informações extraídas por meio da API do YouTube. Essa estrutura modular e distribuída permite que o sistema seja escalável, garantindo flexibilidade para futuras integrações e adaptações de acordo com as necessidades do projeto.

Já a visão lógica apresentada na Figura 5 descreve uma arquitetura dividida em duas camadas principais: a Camada de Aplicação e a Camada de Persistência, com o objetivo de organizar o fluxo de dados desde a coleta inicial até o armazenamento final.

Na Camada de Aplicação, destacam-se os processos de alto nível que guiam a funcionalidade do sistema. O *Web Crawler* é o componente central desta camada, encarregado de coletar dados diretamente das agências verificadoras e da API do YouTube. Esses dados, uma vez obtidos, são transformados em metadados estruturados, representando informações como títulos, datas, rótulos e textos completos. Esses metadados passam então pelo Processamento de Dados, onde são limpos, normalizados e preparados para análises mais avançadas, como a Análise de Sentimento, que interpreta e classifica os dados com base em polaridade ou relevância contextual.

Já a Camada de Persistência foca na operacionalização da coleta e no armazenamento dos dados. Aqui, os *Spiders* das Agências Verificadoras e os *Scripts* das APIs do YouTube são os responsáveis pela execução específica de cada tarefa de coleta. Essas tarefas são gerenciadas pelo Apache *Airflow*, que, por meio de DAGs, organiza, agenda e monitora os processos de execução. O painel de execução da DAG no *Airflow* permite acompanhar em tempo real o status das tarefas, identificando quais foram concluídas com sucesso e quais apresentaram falhas. Os resultados coletados e processados são, então, armazenados em arquivos JSON, que permitem acessibilidade e organização para análises futuras.

Essa visão lógica ressalta a integração fluida entre a coleta de dados, o processamento e o armazenamento. A separação clara de responsabilidades entre as camadas de aplicação e persistência demonstra uma arquitetura bem estruturada, voltada para a escalabilidade e a modularidade.

VII. Avaliação da implementação da arquitetura

A arquitetura desenvolvida se destaca por sua modularidade, permitindo que cada componente possua responsabilidades bem definidas, e por sua extensibilidade, facilitando a adição de novas fontes de dados.

No contexto da arquitetura, a avaliação demonstrou que ela foi funcional e atendeu plenamente às necessidades tanto deste estudo quanto do projeto como um todo. Em relação ao desempenho, a arquitetura demonstrou-se altamente eficiente, sendo capaz de lidar com um grande volume de dados coletados e processados de forma satisfatória. A Figura 6 ilustra os detalhes da execução da DAG, evidenciando que, mesmo com o volume significativo de dados analisados, a conclusão foi alcançada em um tempo de 5 minutos e 38 segundos.

Contudo, no quesito de escalabilidade, identificou-se uma limitação relacionada à extração de dados do YouTube. Esse problema não decorre diretamente da arquitetura proposta, mas sim das restrições impostas pela API do Google, que apresentam limitações no número de requisições e na abrangência das buscas por vídeos. Essas restrições podem resultar em tempos de execução prolongados para buscas que demandam um grande volume de dados, representando um desafio para escalabilidade em cenários mais amplos. Apesar dessa limitação, a arquitetura geral demonstrou-se robusta e eficaz para os propósitos do estudo.

Devido à automação implementada com o *Airflow*, onde o primeiro comando executado é sempre clonar o repositório, a manutenibilidade do sistema é significativamente facilitada. Qualquer alteração realizada no código ou na arquitetura é automaticamente refletida na próxima execução dos *scripts*, eliminando a necessidade de intervenções manuais para atualização das instâncias em execução. Essa abordagem possibilita agilidade na aplicação de mudanças, seja para corrigir falhas, adicionar novas funcionalidades ou ajustar componentes existentes, tornando o processo de manutenção mais eficiente e confiável.

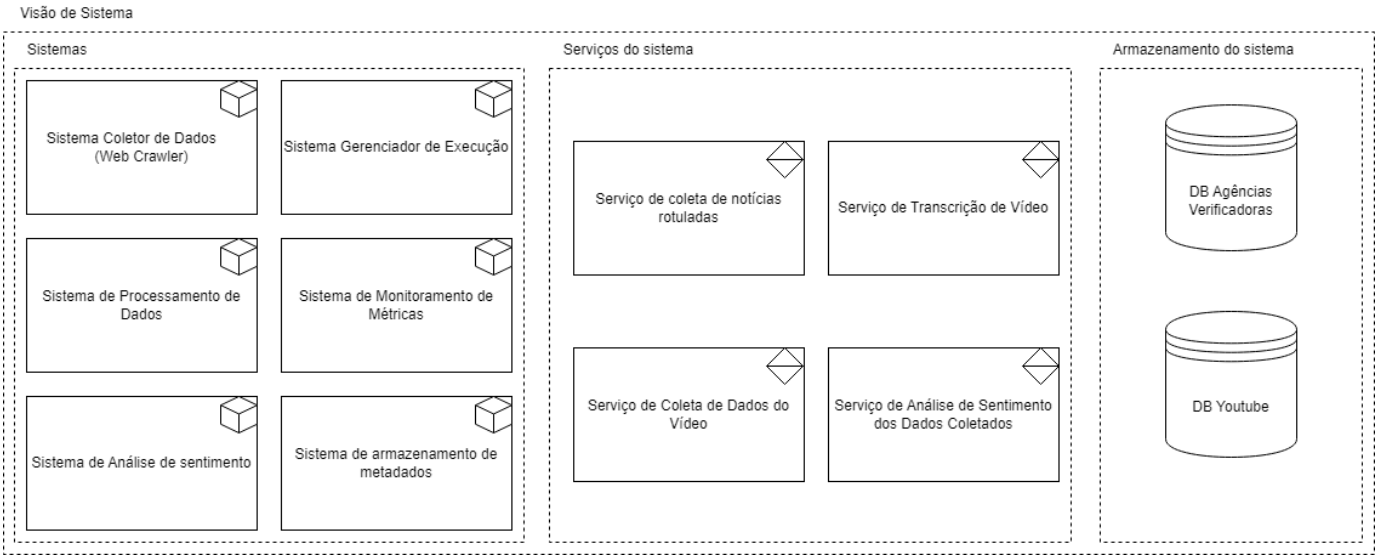

Fig. 4: Visão dos Sistemas - Fonte: Autoria Própria

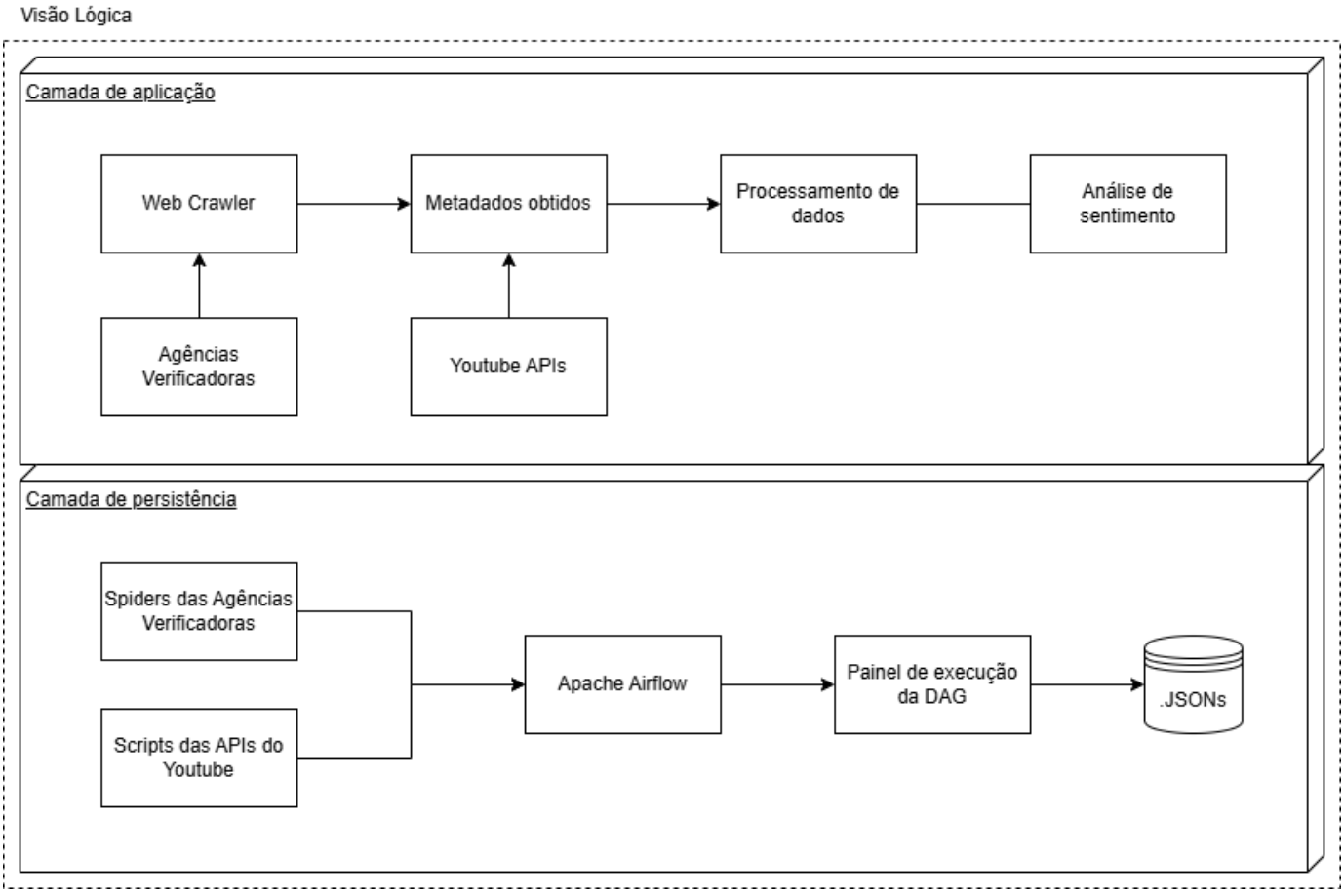

Fig. 5: Visão Lógica - Fonte: Autoria Própria

Na Figura 7, é apresentado um gráfico do histórico de excecuções da DAG criada no *Airflow*, com detalhes sobre os *scripts* concluídos com sucesso e aqueles que apresentaram falhas. Na parte superior, um gráfico de barras exibe a frequência de execuções e o tempo de duração de cada uma, enquanto na parte inferior da imagem estão listados

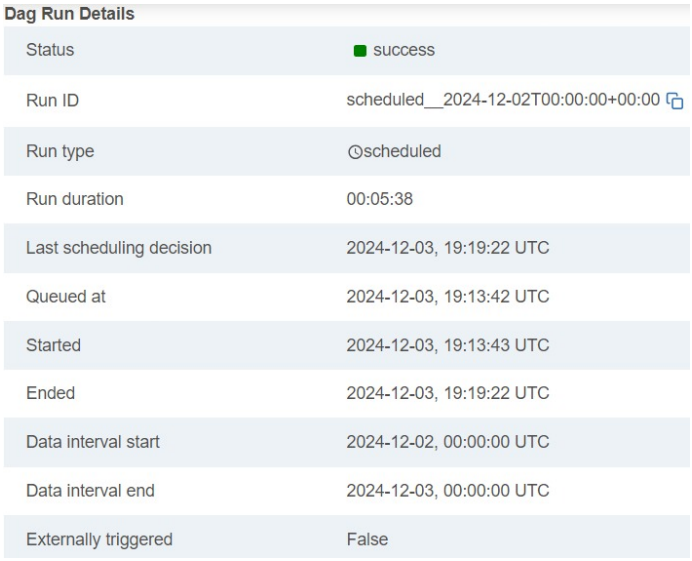

Fig. 6: Painel detalhes DAG do *Airflow* - Fonte: Painel do *Airflow*

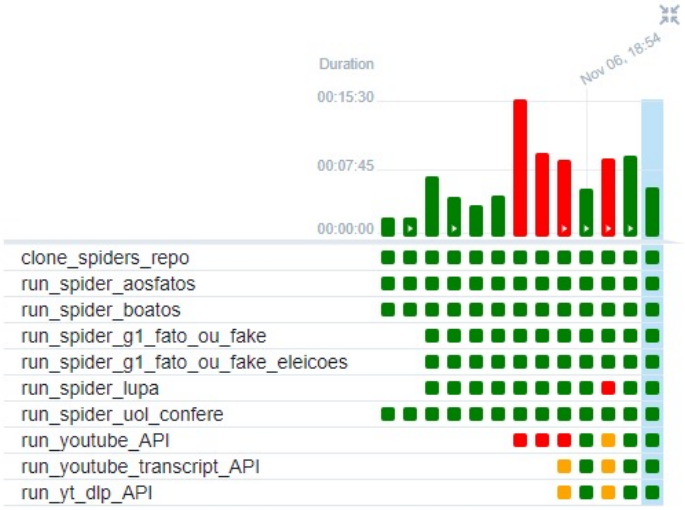

Fig. 7: Painel detalhes DAG do *Airflow* - Fonte: Painel do *Airflow*

todos os *scripts*, juntamente com a ordem em que foram realizados. Essa visualização proporciona um monitoramento claro e eficiente do processo, facilitando a identificação de pontos críticos ou possíveis gargalos no fluxo de trabalho.

VIII. Resultados e Discussões

A. Extração de notícias rotuladas

O extrator desenvolvido, implementado com a biblioteca *Scrapy* e agendado pelo *Airflow*, representa uma ferramenta utilizada para a criação de uma base de dados estruturada a partir de fontes confiáveis de *fact-checking*. As agências verificadoras, como Aos Fatos, Agência Lupa, UOL Confere, G1 Fato ou Fake e Boatos.org, serviram como fontes primárias para a coleta de notícias previamente analisadas por especialistas. Esses dados fornecem um ponto de partida confiável e robusto para o desenvolvimento de soluções de Inteligência Artificial voltadas ao combate à desinformação.

A estruturação dos dados coletados incluiu metadados essenciais, como títulos, textos completos, autores, datas de publicação e rótulos categóricos, como “falso”, “fato” ou “não é bem assim”. Esses rótulos, definidos por especialistas com base em metodologias rigorosas, garantem que o conjunto de dados seja não apenas rico em informações, mas também representativo da realidade. Essa característica é fundamental para o treinamento de modelos de linguagem de grande escala (LLMs), que necessitam de exemplos variados e bem rotulados para aprender a identificar padrões semânticos e contextuais associados à veracidade das informações.

Além disso, o sistema desenvolvido não apenas cria uma base de dados, mas também possibilita o treinamento de LLMs para atribuir scores probabilísticos a notícias não rotuladas. Esses scores indicam a probabilidade de uma notícia ser verdadeira ou falsa, oferecendo uma ferramenta prática para auxiliar os especialistas em *fact-checking*. Dessa forma, o modelo pode priorizar automaticamente as notícias que apresentam maior probabilidade de conter desinformação, otimizando o trabalho manual e aumentando a eficiência das equipes de verificação.

A combinação de fontes confiáveis, um *pipeline* de coleta bem estruturado e a aplicação de técnicas avançadas de aprendizado de máquina destacam o potencial deste extrator. Ele não apenas contribui para a criação de um *dataset* robusto, mas também demonstra como a integração entre tecnologia e expertise humana pode ser utilizada para enfrentar um dos maiores desafios da era da informação: o combate à desinformação de maneira eficiente e escalável.

1) Aos Fatos: O Aos Fatos é uma organização jornalística fundada em 7 de julho de 2015, dedicada ao combate à desinformação, cobertura da tecnopolítica e checagem de fatos. Sediada no Rio de Janeiro, conta com profissionais distribuídos em três regiões do Brasil e dois continentes, aliando jornalismo e tecnologia para promover a integridade da informação, fortalecer políticas públicas e qualificar o debate público.

A missão do Aos Fatos é preservar a integridade da informação no ambiente digital e aumentar o custo da mentira na política, sempre pautado pela ética, transparência e compromisso com a democracia. A organização é signatária do código de conduta da IFCN (*International Fact-Checking Network*) e passa por auditorias anuais que atestam sua credibilidade.

Seu financiamento provém de licenciamento de conteúdo, serviços tecnológicos, patrocínios e o programa de membros Aos Fatos Mais. Não utiliza publicidade programática nem recebe recursos de políticos, campanhas ou governos.

Entre suas iniciativas, destacam-se ferramentas como o Radar Aos Fatos, o *chatbot* Fátima e o transcritor automático Escriba, além de parcerias com plataformas como Facebook, Instagram, WhatsApp, Telegram e Kwai no combate à desinformação. O Aos Fatos também licencia checagens para

empresas e instituições, promovendo políticas de informação confiáveis.

Reconhecido nacional e internacionalmente, recebeu prêmios como o Gabriel García Márquez 2020 e os Digital Media Américas 2024 e 2023. Todo o conteúdo produzido é protegido por direitos autorais e distribuído em diversas plataformas, como YouTube, TikTok e LinkedIn.

A Tabela I apresenta os metadados obtidos da Agência Aos Fatos, de modo a exemplificar e descrever como foi realizada a raspagem de dados. Nesse site, além das informações comuns em verificações existem também os campos de texto "*fact*" onde a notícia é destacada e "*check*", no qual o especialista explica como foi realizada a checagem da informação, como é exemplificado no Apêndice A, Figura 10.

TABELA I: Metadados de notícias da base "Aos Fatos"

Metadados	Descrição	Tipo
link	Link da notícia	String
title	Título da notícia	String
data	Data da notícia	String
author	Autor da notícia	String
text	Texto da notícia na íntegra	String
fact	Fato a ser checado	String
check	Checagem do fato	String
tag	Rótulo definido por especialistas	String

2) Boatos: O Boatos.org foi criado em junho de 2013 pelo jornalista Edgard Matsuki com o objetivo de combater a desinformação nas redes sociais. A iniciativa surgiu da vontade de inovar no jornalismo e de combater o crescente conteúdo falso nas redes sociais. Com o tempo, o site se expandiu, tornando-se um dos maiores portais de *fact-checking* do Brasil, com uma média de 100 publicações mensais e mais de 1 milhão de acessos mensais. O Boatos.org também atua como ferramenta de educação midiática, ensinando como identificar e checar notícias falsas.

A metodologia do Boatos.org envolve a verificação de conteúdos virais por meio de análise de dados, consulta a fontes primárias e secundárias, além de entrevistas. A partir de 2023, o site passou a classificar os conteúdos em diferentes categorias, como "*Fake news*", "Golpe", "Verdadeiro" e "Em apuração", para tornar mais claro o status das checagens. A escolha das pautas é feita com base em sugestões de leitores, volume de buscas e monitoramento das redes sociais, sem critérios subjetivos.

O site é mantido por uma equipe pequena, composta por dois jornalistas, e seu financiamento provém principalmente de publicidade programática, com apoio de parcerias pontuais com organizações como o Portal Metrópoles e entidades como o CNJ e TSE. O Boatos.org se mantém independente, sem receber fundos públicos, e continua sendo uma referência para a checagem de fatos no Brasil.

A Tabela II apresenta os metadados obtidos da Agência Boatos, de modo a exemplificar e descrever como foi realizada a raspagem de dados. Nesse site, foram extraídas apenas as informações comuns em verificações, porém, é válido ressaltar que, como é exemplificado no Apêndice, A Figura 11, existe uma maior variedade de rótulos usados no campo de texto "tag", utilizando até mesmo emojis para enfatizar o veredito dos especialistas.

TABELA II: Metadados de notícias da base "Boatos"

Metadados	Descrição	Tipo
link	Link da notícia	String
title	Título da notícia	String
data	Data da notícia	String
author	Autor da notícia	String
text	Texto da notícia na íntegra	String
tag	Rótulo definido por especialistas	String

3) G1 Fato ou Fake: O Fato ou Fake, lançado em 2018 pelo Grupo Globo, é um serviço dedicado à checagem de conteúdos duvidosos amplamente compartilhados na internet. Com a participação de jornalistas de veículos como G1, O Globo, Extra, Época, Valor, CBN, GloboNews e TV Globo, a equipe realiza monitoramento diário para verificar mensagens suspeitas em redes sociais e aplicativos como WhatsApp. Em mais de quatro anos de atuação, já foram realizadas mais de 3 mil checagens, abrangendo boatos e declarações de políticos, especialmente durante períodos eleitorais.

O serviço utiliza fontes oficiais e especialistas para garantir a precisão das verificações. Os resultados das checagens são divulgados nos perfis do Fato ou Fake no Facebook, Twitter e Instagram, promovendo acessibilidade e transparência. Além disso, os usuários podem sugerir conteúdos para checagem diretamente à equipe por meio do WhatsApp.

Combinando a expertise de várias redações e uma metodologia ágil, o Fato ou Fake desempenha um papel essencial na luta contra a desinformação no Brasil, ajudando a distinguir notícias verdadeiras de conteúdos falsos e fortalecendo a confiança pública na informação.

A Tabela III apresenta os metadados obtidos do portal de notícias G1, de modo a exemplificar e descrever como foi realizada a raspagem de dados. Nesse site, foram extraídas apenas as informações comuns em verificações, porém, nessa abordagem são utilizados vários rótulos para diferentes informações no corpo do texto de verificação, assim como está exemplificado no Apêndice A, Figura 12.

TABELA III: Metadados de notícias das bases "G1 Fato ou Fake"

Metadados	Descrição	Tipo
link	Link da notícia	String
title	Título da notícia	String
data	Data da notícia	String
author	Autor da notícia	String
text	Texto da notícia na íntegra	String
tag	Rótulo definido por especialistas	String

4) Lupa: A Lupa é uma agência de notícias fundada em 2015, pioneira no Brasil em *fact-checking* e referência no combate à desinformação. Com um compromisso com a análise de dados e a transparência, tornou-se a primeira plataforma especializada no tema a integrar o *The Trust Project*, em 2019, destacando-se entre as iniciativas globais de mídia transparente e acessível. Também é signatária da *International Fact-Checking Network* (IFCN), seguindo rigorosamente seus

princípios éticos, incluindo o apartidarismo, adotado por todos os seus colaboradores.

A atuação da Lupa se divide em duas frentes principais: Lupa Jornalismo, focada em checagens, reportagens e verificações, e Lupa Educação, que promove oficinas, treinamentos e projetos de educação midiática em escolas, universidades e instituições. Por meio dessas iniciativas, busca sensibilizar a sociedade sobre os riscos da desinformação e capacitar indivíduos para enfrentá-la de forma crítica. Além disso, mantém parcerias em projetos especiais voltados à produção de conteúdo jornalístico e à expansão da discussão sobre o impacto da desinformação na democracia.

A Tabela IV apresenta os metadados obtidos da Agência Lupa, de modo a exemplificar e descrever como foi realizada a raspagem de dados. Nesse site, além das informações comuns em verificações existe também o campo de texto "*subject*" onde é informado o tema ao qual a notícia é relacionada, como é exemplificado no Apêndice, A Figura 13.

TABELA IV: Metadados de notícias da base "Lupa"

Metadados	Descrição	Tipo
link	Link da notícia	String
title	Título da notícia	String
data	Data da notícia	String
author	Autor da notícia	String
text	Texto da notícia na íntegra	String
tag	Rótulo definido por especialistas	String
subject	Tema da notícia	String

5) UOL confere: O UOL Confere é a divisão do UOL dedicada à checagem e esclarecimento de fatos. Fundada com base nos Princípios Editoriais do Manual de Redação da Folha e no Código de Princípios da *International Fact-Checking Network* (IFCN), a equipe do UOL Confere segue parâmetros rigorosos, como independência, pluralismo, apartidarismo, interesse público e transparência. Isso inclui a clareza sobre as fontes consultadas, o financiamento da organização e a equipe envolvida na checagem.

O método de checagem do UOL Confere é estruturado para garantir a veracidade das informações e combate à desinformação. Além disso, o processo é orientado pela transparência, permitindo que os leitores acompanhem os detalhes das fontes e da metodologia utilizada. O UOL Confere tem como objetivo garantir informações confiáveis, contribuindo para um ambiente de comunicação mais preciso e democrático.

A Tabela V apresenta os metadados obtidos da Agência Lupa, de modo a exemplificar e descrever como foi realizada a raspagem de dados. Nesse site, não foi possível extrair os rótulos de cada notícia. Tal falha se deve ao fato de que a estrutura HTML do site não padroniza um campo para o rótulo, além disso os especialistas dissertam sobre o veredito de forma menos objetiva em relação às outras agências, evitando o uso de rótulos, como é exemplificado no Apêndice A, Figura 14.

TABELA V: Metadados de notícias da base "UOL"

Metadados	Descrição	Tipo
link	Link da notícia	String
title	Título da notícia	String
data	Data da notícia	String
author	Autor da notícia	String
text	Texto da notícia na íntegra	String

B. Extração de vídeos do YouTube

1) YouTube Data API v3: A API YouTube Data v3 é uma ferramenta que permite interagir com os vastos dados disponíveis na plataforma do YouTube de forma programática, oferecendo uma maneira eficiente de acessar, gerenciar e explorar conteúdo. Essa API é amplamente utilizada para tarefas como busca de vídeos, obtenção de informações detalhadas sobre canais, *playlists*, estatísticas de vídeos, e muito mais. A funcionalidade de busca é um dos recursos mais importantes e flexíveis da API, permitindo que desenvolvedores realizem consultas precisas para localizar vídeos, canais ou *playlists* com base em critérios variados, como palavras-chave, categorias, datas de publicação e relevância.

A seção de busca funciona com base em parâmetros específicos que podem ser configurados na requisição. Por exemplo, o parâmetro 'q' é usado para definir as palavras-chave que serão pesquisadas. Outros parâmetros, como '*type*', podem restringir os resultados a vídeos, canais ou *playlists*. A pesquisa também pode ser refinada usando opções como '*order*', que determina a ordenação dos resultados, seja por relevância, data de publicação, contagem de visualizações ou avaliações. Além disso, filtros como '*publishedAfter*' e '*publishedBefore*' permitem delimitar um intervalo de tempo para os resultados, enquanto o parâmetro '*regionCode*' possibilita buscar conteúdo relacionado a um país específico.

Utilizando o *endpoint* de pesquisa, ao utilizar as funções apresentadas no Apêndice B, Código 2 foi possível configurar requisições que retornassem resultados diversificados e relevantes, abrangendo títulos, descrições, IDs de vídeos, datas de publicação e canais associados. Esses metadados fornecem uma base rica para análises posteriores, seja no âmbito de estudos acadêmicos, pesquisas de mercado ou desenvolvimento de aplicações baseadas em dados.

No entanto, é importante considerar as limitações impostas pela API para garantir seu uso eficiente. Cada desenvolvedor recebe uma cota diária de 10.000 unidades de consulta (quota), com cada tipo de requisição consumindo uma quantidade específica dessa cota. No caso da funcionalidade de busca, cada requisição consome 100 unidades, o que significa que é possível realizar até 100 buscas completas por dia no período gratuito. A consulta pode ser otimizada ajustando os parâmetros e refinando os filtros para maximizar a relevância dos resultados retornados.

O uso gratuito da API está vinculado a um cadastro inicial no *Google Cloud Platform*, que oferece um crédito promocional para novos usuários. Esse crédito permite experimentar a API sem custos iniciais, desde que o consumo permaneça dentro do limite de cota diário e do crédito disponível. Essa

política é especialmente vantajosa para desenvolvedores em fase de teste ou pequenas implementações. Contudo, para projetos maiores, que demandam uma quantidade maior de consultas diárias ou operações mais intensivas, é necessário considerar a possibilidade de custos adicionais, adquirindo cotas extras conforme necessário.

Ao integrar a API ao trabalho, o planejamento cuidadoso das requisições e o monitoramento constante do consumo de cota foram fundamentais para garantir a eficiência e continuidade do projeto. Essa abordagem não só permitiu obter uma ampla gama de metadados de vídeos, mas também assegurou que o uso da API estivesse em conformidade com suas limitações técnicas e financeiras. A API YouTube Data v3, com sua funcionalidade robusta e detalhada de busca, provou ser uma ferramenta indispensável para atingir os objetivos do projeto, destacando-se como um recurso versátil e poderoso para explorar o vasto universo de conteúdo do YouTube.

A tabela VI detalha os metadados extraídos dos vídeos por meio da API YouTube Data v3, organizados em três categorias principais: dados relacionados ao vídeo e ao canal, estatísticas do vídeo e informações de comentários. Os campos incluem identificadores como '*videoId*' e '*channelId*', fundamentais para localizar vídeos e canais na plataforma, além de dados contextuais como o título '*title*', descrição '*description*' e a data de publicação '*publishedAt*'. Esses elementos são armazenados como strings, facilitando o uso e a manipulação em sistemas externos. Na seção de estatísticas, informações como número de visualizações '*viewCount*', curtidas '*likeCount*' e comentários '*commentCount*' fornecem uma visão geral da popularidade e do engajamento do vídeo. A última seção, que cobre os comentários, apresenta informações detalhadas como o autor do comentário '*authorDisplayName*', o texto '*textDisplay*' e o número de curtidas do comentário '*likeCount*', permitindo uma análise qualitativa da interação dos usuários.

TABELA VI: Metadados dos Vídeos, Estatísticas e Comentários

Metadados	**Descrição**	**Tipo**
Dados relacionados ao vídeo e ao canal		
videoId	ID do vídeo, também é o final do URL	String
publishedAt	Data de publicação do vídeo	String
channelId	ID do canal	String
title	Título do vídeo	String
description	Descrição do vídeo	String
channelTitle	Nome do canal	String
Dados estatísticos do vídeo		
viewCount	Número de visualizações do vídeo	String
likeCount	Número de curtidas do vídeo	String
favoriteCount	Número de favoritações do vídeo	String
commentCount	Número de comentários do vídeo	String
Dados de todos os comentários do vídeo		
authorDisplayName	Nome do canal que comentou	String
textDisplay	Conteúdo do comentário	String
likeCount	Número de curtidas do comentário	String

O arquivo JSON, resultante do trabalho de coleta confirma o sucesso da aplicação desses parâmetros e técnicas. Nele, os dados aparecem organizados de maneira clara, refletindo as categorias descritas na tabela. Por exemplo, no caso de vídeos relacionados às eleições municipais de Goiânia, é possível verificar uma grande riqueza de informações, desde metadados como o título até a lista de comentários detalhados, contendo opiniões diversas sobre os candidatos. Esse nível de detalhamento é especialmente útil para estudos mais aprofundados, como análises de opinião pública ou de popularidade de conteúdos, demonstrando o alcance e a relevância do uso da API YouTube Data v3.

2) YouTube Transcript API: A *API YouTube Transcript* é uma ferramenta útil para a obtenção de transcrições de vídeos no YouTube, permitindo acessar o conteúdo textual falado nos vídeos de forma programática. Essa API, disponível como um pacote *Python*, é capaz de extrair as transcrições de vídeos que possuem legendas automáticas ou manuais habilitadas, fornecendo as falas organizadas cronologicamente junto com os respectivos *timestamps*. Isso possibilita o uso de transcrições em diversas aplicações, como análises de discurso, criação de legendas personalizadas, mineração de dados ou indexação de conteúdos. O funcionamento da API é baseado no envio do identificador único do vídeo, o '*videoId*', que foi obtido previamente por meio da API YouTube Data v3. Uma vez fornecido o '*videoId*', a API busca a transcrição, retornando os dados estruturados em um formato JSON, facilitando a integração com outros processos e sistemas.

Integrada ao fluxo de trabalho, iniciado com a coleta de IDs de vídeos via API YouTube Data v3, a ferramenta possibilitou a obtenção automática das transcrições, a partir da função apresentada no Apêndice B, Código 3. Isso permitiu que o conteúdo de vídeos variados fosse convertido para texto, viabilizando análises aprofundadas e aplicações que demandam processamento textual. Por exemplo, os dados transcritos poderiam ser usados em algoritmos de processamento de linguagem natural para detectar padrões de discurso, sentimentos ou tendências, ou ainda em sistemas de busca que indexam o conteúdo falado nos vídeos.

3) yt-dlp API: Uma ferramenta de linha de comando, projetada para realizar o download de vídeos e áudios do YouTube. Sua flexibilidade e robustez permitem que desenvolvedores extraiam conteúdo de vídeo em diferentes formatos e qualidades, adaptando o processo às necessidades específicas de cada projeto. Para operar, a *yt-dlp API* utiliza o identificador único de vídeo, o '*videoId*', que pode ser fornecido diretamente ou obtido previamente por meio da API YouTube Data v3. Com esse identificador, a ferramenta é capaz de localizar o conteúdo na plataforma e baixá-lo, respeitando as configurações definidas pelo usuário, como qualidade do vídeo, formato do arquivo e inclusão de metadados.

A partir da execução da função apresentada no Apêndice B, Código 4, a API provou-se extremamente eficiente, permitindo configurar opções detalhadas, como a resolução dos vídeos, o formato de saída e a inclusão de legendas embutidas. Esses vídeos baixados foram armazenados localmente, possibilitando análises offline e manipulação posterior, como processamento de imagem, extração de áudio ou utilização em sistemas de aprendizado de máquina.

Para o funcionamento completo e eficiente da *yt-dlp API*, é necessário o software *FFmpeg* instalado no sistema. O *FFmpeg* é um conjunto de bibliotecas que permite a manipulação de arquivos de mídia, sendo essencial para operações como a conversão de formatos, extração de áudio e combinação de fluxos de vídeo e legendas.

No entanto, é importante observar que o uso d *yt-dlp API* para o download de vídeos do YouTube pode violar os Termos de Serviço da plataforma. O YouTube especifica que os usuários não devem baixar vídeos, exceto quando explicitamente permitido, como em vídeos disponibilizados para download por meio do botão oficial. Portanto, o uso dessa ferramenta para finalidades que contrariam as diretrizes da plataforma pode levar a consequências legais ou ao bloqueio do acesso.

C. Análise de sentimento

1) Comentários: A análise de sentimento dos comentários capturados pela API está apresentada no Apêndice C, Figura 15. A análise de sentimento foi realizada utilizando a biblioteca "*transformers*" e o modelo público "*nlptown/bert-base-multilingual-uncased-sentiment*", disponibilizado pela plataforma *Hugging Face* e especializado em classificar sentimentos em textos. Inicialmente, os dados foram carregados de um arquivo JSON, contendo informações de vídeos relacionados às Eleições de Goiânia 2024 e seus respectivos comentários. O modelo foi configurado para processar os textos de forma eficiente, permitindo a tokenização e a inferência nos comentários.

Para cada comentário, o texto foi tokenizado, garantindo o ajuste necessário para entrada no modelo, com truncamento e preenchimento automático para um comprimento máximo de 128 tokens. Após a tokenização, foi realizada a inferência no modelo, que gerou logits, valores utilizados para calcular a probabilidade de cada rótulo de sentimento. A classificação final foi obtida identificando o índice do logit mais alto, correspondente ao sentimento predominante no comentário.

Os rótulos de sentimentos foram mapeados para categorias compreensíveis: "muito negativo", "negativo", "neutro", "positivo" e "muito positivo". Cada vídeo foi processado individualmente, verificando-se se havia comentários associados. Para os vídeos com comentários, foi feita a análise de sentimento em lote, armazenando os resultados em uma estrutura que associava o ID do vídeo aos seus comentários e respectivos sentimentos analisados.

Por fim, os resultados da análise foram estruturados e salvos em um arquivo JSON denominado *"analise_sentimento_comentarios.json"*, contendo os identificadores dos vídeos, os textos dos comentários e os sentimentos atribuídos. Este processo garantiu uma análise eficiente e detalhada dos sentimentos expressos nos comentários, utilizando um modelo robusto da Hugging Face, com resultados prontos para consultas ou análises posteriores.

Observou-se que o modelo frequentemente classificou incorretamente comentários, identificando erros nos sentimentos positivos como negativos. Essa imprecisão evidencia que o BERTopic não seria a escolha mais adequada para este tipo de análise de sentimento.

Adicionalmente, utilizou-se a biblioteca VADER para análise de sentimento, cujos resultados revelaram-se ainda mais comprometidos que os do BERTopic. O modelo demonstrou desempenho extremamente limitado, praticamente não conseguindo identificar ou extrair qualquer valor significativo dos comentários. Essa ineficácia reforça a necessidade de desenvolver ou selecionar métodos mais robustos de análise de sentimento para este tipo de dados.

2) Transcrições: Alterando apenas o vdataset a ser utilizado, a implementação da análise usando o BERTopic foi bastante semelhante, porém para as transcrições, a análise de sentimento não se mostrou pertinente devido à natureza do conteúdo veiculado em vídeo. Tratando-se de um formato noticioso, a extração de sentimentos não apresenta relevância metodológica significativa. Os textos originados de reportagens jornalísticas tendem a manter um padrão de neutralidade e objetividade, priorizando a transmissão de informações sobre a manifestação de emoções ou opiniões pessoais.

D. Integração com o projeto dAurora

A etapa de integração com a DAurora consistiu no envio dos dados coletados e processados, Figura 9, para o DAurora por meio de requisições POST realizadas através da API do próprio projeto. Essa abordagem foi escolhida devido à automação do sistema via *Airflow*, garantindo que os dados fossem transmitidos para o *frontend* do DAurora e cadastrados como uma postagem como é apresentado na Figura 8. No entanto, como o projeto ainda está em desenvolvimento, essa solução foi adotada como uma alternativa temporária, e a definição de uma abordagem mais robusta e adequada para a transferência de dados será realizada em etapas futuras, conforme o progresso do projeto e suas necessidades específicas.

IX. Limitações Identificadas

A. Futuras alterações nas estruturas dos sites

A implementação dos *crawlers* depende fortemente das estruturas HTML das páginas web das agências de *fact-checking* e de outras plataformas utilizadas para coleta de dados. Alterações nas estruturas desses sites, como mudanças nos seletores de classes, atributos ou hierarquias do código HTML, podem comprometer a funcionalidade dos *spiders* desenvolvidos, levando à necessidade de ajustes constantes nos códigos. Por exemplo, foi relatado durante o desenvolvimento do projeto que algumas agências utilizam carregamento dinâmico via *JavaScript*, o que requer ferramentas como Playwright para renderização do conteúdo antes da extração. Caso novos mecanismos de bloqueio sejam implementados ou mudanças significativas ocorram, o processo de coleta de dados pode ser inviabilizado temporariamente, impactando diretamente na validade e na continuidade do estudo.

B. Não padronização dos rótulos

Uma das principais limitações encontradas está relacionada à diversidade de rótulos utilizados pelas agências de *fact-checking* para classificar as notícias verificadas. Cada agência

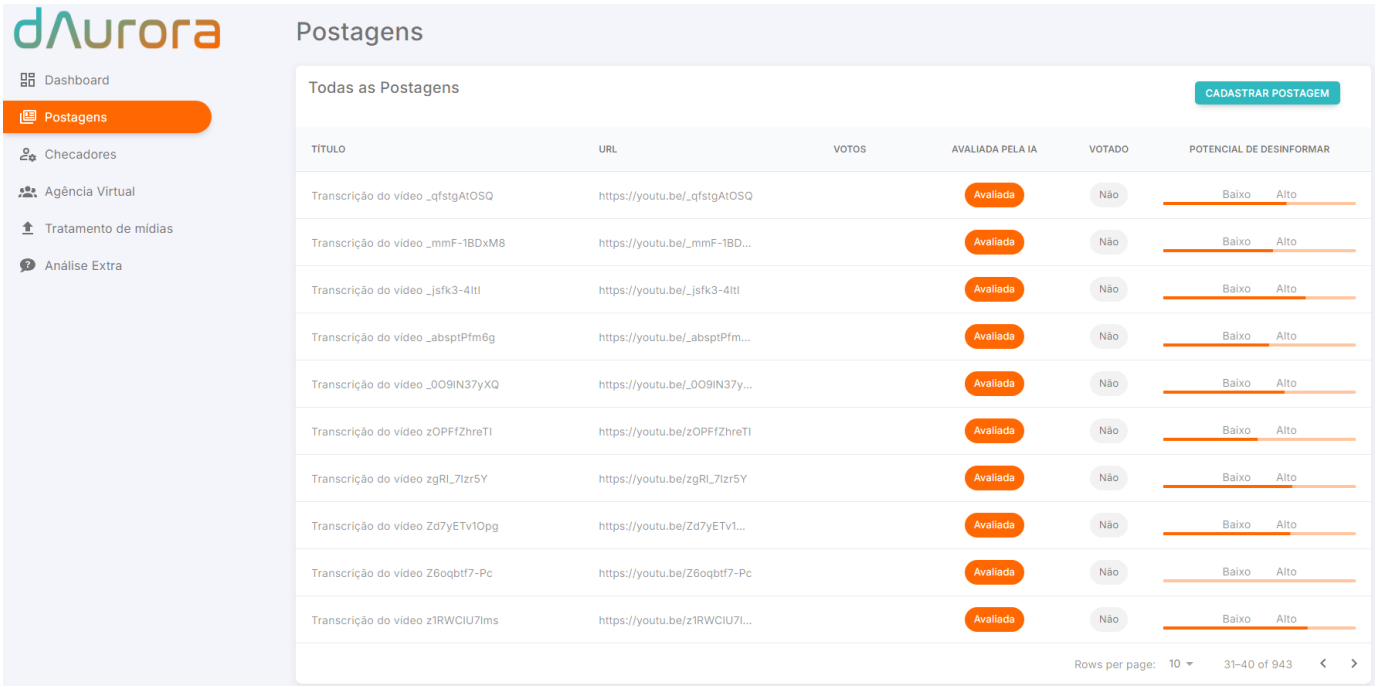

Fig. 8: Tela com as postagens extraídas de vídeos do Youtube

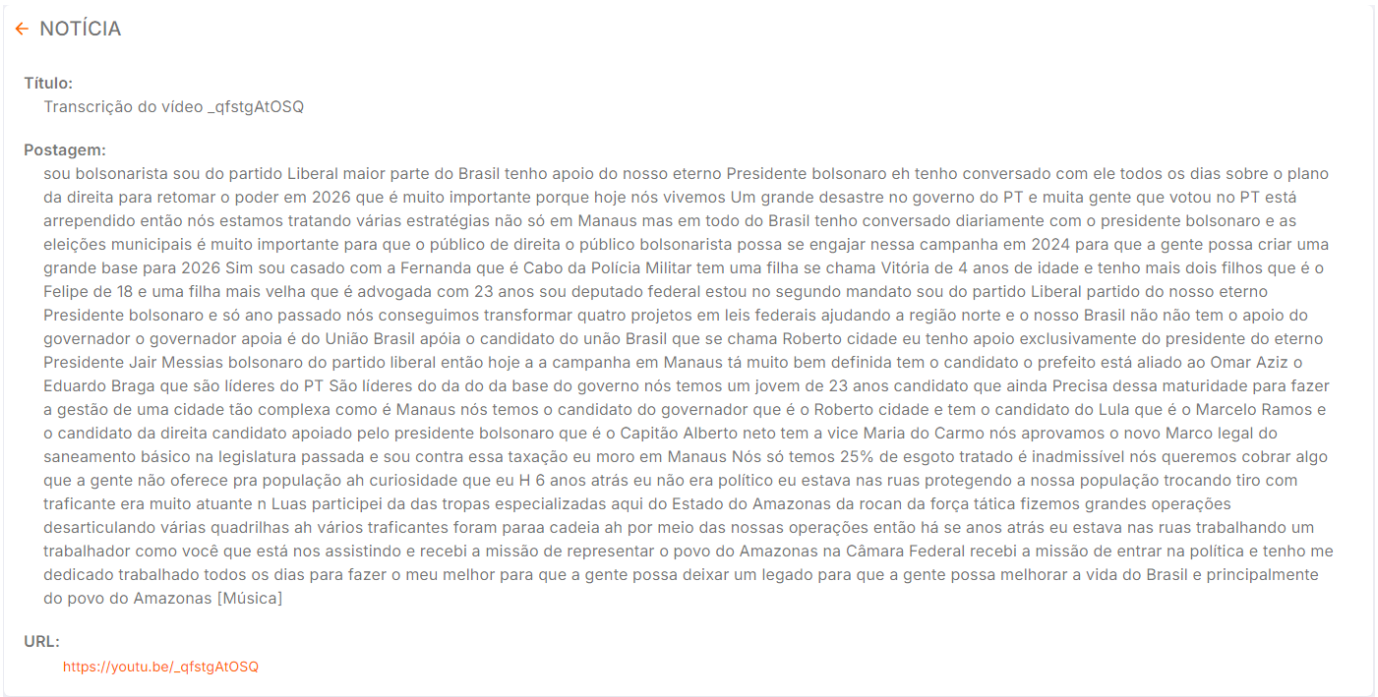

Fig. 9: Exemplo dos dados de uma postagem extraída do Youtube

adota terminologias distintas, como “falso”, “impreciso”, “meia-verdade” ou “verdadeiro”, além de utilizar elementos visuais como emojis para reforçar suas classificações. Essa falta de padronização dificulta a integração semântica dos dados, exigindo uma interpretação cuidadosa e a definição de uma taxonomia consolidada no ecossistema do Projeto Web 3.0. Embora os dados tenham sido coletados e mantidos em seu formato original para preservar a riqueza informacional, futuras análises podem ser comprometidas caso não seja implementado um mecanismo padronizado de conversão desses rótulos.

C. Esgotamento das cotas da API do Youtube

A coleta de dados via YouTube Data API v3 é limitada por cotas de uso que restringem o número de requisições diárias e a profundidade das buscas realizadas. Durante o desenvolvimento do estudo, as buscas foram configuradas de forma a otimizar o uso dos recursos disponíveis, mas em cenários que demandem uma coleta em larga escala, há o risco de esgotamento das cotas. Caso isso ocorra, o processo de coleta será interrompido, exigindo a aquisição de mais cotas, seja por meio de solicitação ao Google ou pela implementação de contas adicionais. Essa limitação pode impactar diretamente na capacidade de coletar dados de forma consistente e tempestiva, especialmente em contextos de alta demanda, como eleições ou crises informacionais.

D. Limitações impostas pelos Termos de Serviço do Youtube

Os Termos de Serviço do YouTube impõem restrições significativas no uso de dados coletados pela API, proibindo, por exemplo, o armazenamento de vídeos ou o uso de conteúdos de forma que viole os direitos autorais. Apesar de ter sido testada a possibilidade de download de vídeos utilizando a *yt-dlp API*, essa abordagem foi descartada no âmbito do Projeto Web 3.0 devido às diretrizes de uso da plataforma. Tais restrições podem limitar a profundidade das análises realizadas e a disponibilidade de dados para investigações futuras, exigindo que todas as operações estejam em conformidade com as políticas estabelecidas pela plataforma. Qualquer descumprimento dessas regras pode resultar na suspensão do acesso à API e na perda de dados coletados, comprometendo a validade e a continuidade do estudo.

X. Conclusão

Este estudo reafirma a relevância e a necessidade de soluções tecnológicas inovadoras para enfrentar os desafios impostos pela desinformação nas plataformas digitais. Por meio do desenvolvimento de uma arquitetura de software robusta e integrada, o trabalho apresentou avanços significativos na coleta, processamento e organização de dados provenientes de fontes confiáveis nas agências de *fact-checking*, e redes sociais digitais como fontes complementares de dados, com destaque para a utilização da API do YouTube.

A combinação de ferramentas como *Scrapy*, *Playwright* e *Apache Airflow* possibilitou a criação de um *pipeline* eficiente e escalável para coleta automatizada de dados, atendendo às exigências do ecossistema do Projeto DAurora. Além disso, a integração de métodos de análise de sentimento, utilizando modelos como VADER e BERTopic, proporcionou análises sobre as dinâmicas da desinformação e os impactos sociais decorrentes.

Os resultados obtidos demonstram que a metodologia proposta não só supera os desafios técnicos associados à coleta e à padronização de dados, mas também estabelece uma base sólida para futuras investigações científicas e aplicações práticas no combate à desinformação. A possibilidade de estruturar dados de forma consistente e de automatizar processos que antes dependiam exclusivamente de esforços manuais representa um passo importante para aumentar a eficiência e a abrangência das iniciativas de verificadores de fatos e pesquisadores.

Entretanto, também foram identificadas limitações importantes, como as restrições impostas pela API do YouTube em relação ao número de requisições e à abrangência das buscas. Essas barreiras reforçam a necessidade de exploração de soluções complementares e o aprimoramento contínuo da arquitetura desenvolvida para lidar com cenários mais complexos e escaláveis.

No âmbito acadêmico e prático, este trabalho contribui com um modelo metodológico que alia expertise humana e tecnologia para abordar um dos maiores desafios da sociedade contemporânea: a fragmentação do debate público causada pela desinformação. Espera-se que a solução proposta inspire novos estudos e o desenvolvimento de iniciativas que promovam a transparência e a confiabilidade das informações disseminadas no ambiente digital, fortalecendo assim as bases de uma sociedade mais informada e resiliente.

Para trabalhos futuros, é possível aprimorar o sistema para lidar com a duplicação de notícias, um problema recorrente que pode levar ao preenchimento excessivo do banco de dados com informações redundantes. A implementação de métodos eficientes de detecção e filtragem de conteúdos duplicados é essencial para otimizar o armazenamento e a qualidade dos dados.

Também é viável ampliar o escopo de coleta de dados, incluindo portais de notícias que vão além das agências de *fact-checking*, além de explorar outras redes sociais. Essa expansão permitiria diversificar e enriquecer as fontes de dados, fortalecendo ainda mais as iniciativas no combate à disseminação de notícias falsas, ao oferecer uma visão mais ampla e abrangente do ecossistema de desinformação.

Além do *crawler*, podem ser desenvolvidos métodos para analisar como conteúdos desinformativos se propagam nas redes sociais. Isso poderia incluir a identificação de padrões de disseminação, análise de redes para mapear conexões e fluxos de informação, e a detecção de influenciadores ou grupos que desempenham papéis centrais na amplificação de *fake news*. Essas abordagens ampliariam a compreensão do fenômeno e ofereceriam subsídios valiosos para a elaboração de intervenções mais direcionadas e eficazes.

Agradecimentos

Aos nossos colegas de classe, registramos nossa profunda gratidão pela parceria, apoio e colaboração ao longo desses anos. Cada um de vocês foi fundamental para tornar essa jornada acadêmica mais rica e significativa.

Aos professores que, com dedicação, paciência e compromisso, compartilharam seus conhecimentos e orientações, nossos sinceros agradecimentos. Suas contribuições foram essenciais para nosso crescimento pessoal e profissional.

Estendemos também nossa gratidão a todos os profissionais da Universidade Federal de Goiás, cujas diversas áreas e esforços tornam possível a formação de tantos alunos,

incluindo a nossa. Obrigado por sua dedicação e trabalho contínuo.

Ao Professor Jacson Rodrigues Barbosa, expressamos nosso mais sincero agradecimento por nos proporcionar a oportunidade de realizar este estudo. Ao Valdemar Vicente Graciano Neto e ao Eliomar Araújo de Lima, nossa gratidão por nos acolherem e confiarem em nosso trabalho no Projeto Web 3.0.

Referências

[1] Sônia Cristina Vermelho et al. "Refletindo sobre as redes sociais digitais". In: *Educação & sociedade* 35 (2014), pp. 179–196.

[2] Simon Kemp. *Digital 2024: Global Overview Report*. Acessado em: 29 de outubro de 2024. 2024. URL: https://datareportal.com/reports/digital-2024-global-overview-report.

[3] Dan Jones. *How Medieval Fake News Brought Down the Knights Templar*. Acessado em: 29 de outubro de 2024. 2017. URL: https://time.com/4981316/friday-13th-knights-templar-post-truth/.

[4] Alison Flood. *Fake news is 'very real' word of the year for 2017*. Acessado em: 29 de outubro de 2024. 2017. URL: https://www.theguardian.com/books/2017/nov/02/fake-news-is-very-real-word-of-the-year-for-2017.

[5] Academia Brasileira de Letras. *Pós-verdade*. Acessado em: 5 de novembro de 2024. Nov. 2024. URL: https://www.academia.org.br/nossa-lingua/nova-palavra/pos-verdade.

[6] Eliomar Araújo de Lima et al. *Projeto Web 3.0 - Avaliação de Impacto da Web 3.0: Descentralizada, Imersiva, Semântica, Centrada no Usuário e Conectada com o Mundo Ciberfísico; Relatório Técnico - Fake News – Etapa 4 – Relatório 2 – PoC dApp*. Tech. rep. 02-2024. TechReport In Portuguese (Under Development) Restricted Access. Universidade Federal de Goiás, Apr. 2024.

[7] Valdemar Graciano-Neto et al. "Estabelecendo uma Arquitetura Baseada em Blockchain para Detecção de Fake News". In: *Anais do XVIII Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software*. Curitiba/PR: SBC, 2024, pp. 91–100. DOI: 10.5753/sbcars.2024.3899. URL: https://sol.sbc.org.br/index.php/sbcars/article/view/30236.

[8] Guido VanRossum and Fred L Drake. *The python language reference*. Vol. 561. Python Software Foundation Amsterdam, The Netherlands, 2010.

[9] Brian Pinkerton. *Webcrawler: Finding what people want*. University of Washington, 2000.

[10] C.J. Hutto and E.E. Gilbert. "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text". In: *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*. Ann Arbor, MI, June 2014.

[11] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[12] Oracle Corporation. *O que é um banco de dados?* Acessado em: 13 de novembro de 2024. Nov. 2020. URL: https://www.oracle.com/br/database/what-is-database/.

[13] AWS Amazon. *O que é uma API (interface de programação de aplicações)?* Acessado em: 13 de novembro de 2024. 2024. URL: https://aws.amazon.com/pt/what-is/api/.

[14] Xuesong Zhai et al. "A Review of Artificial Intelligence (AI) in Education from 2010 to 2020". In: *Complexity* 2021.1 (2021), p. 8812542. DOI: https://doi.org/10.1155/2021/8812542. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/8812542. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/8812542.

[15] Philippe Kruchten. "Common misconceptions about software architecture". In: *The Rational Edge* 1 (2001), p. 1998.

[16] Ian L Alberts et al. "Large language models (LLM) and ChatGPT: what will the impact on nuclear medicine be?" In: *European journal of nuclear medicine and molecular imaging* 50.6 (2023), pp. 1549–1552.

[17] Jaime G. Carbonell, Ryszard S. Michalski, and Tom M. Mitchell. "1 - AN OVERVIEW OF MACHINE LEARNING". In: *Machine Learning*. Ed. by Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell. San Francisco (CA): Morgan Kaufmann, 1983, pp. 3–23. ISBN: 978-0-08-051054-5. DOI: https://doi.org/10.1016/B978-0-08-051054-5.50005-4. URL: https://www.sciencedirect.com/science/article/pii/B9780080510545500054.

[18] Yasmim Mendes Rocha et al. "The impact of fake news on social media and its influence on health during the COVID-19 pandemic: A systematic review". In: *Journal of Public Health* (2021), pp. 1–10.

[19] Scrapy. *Scrapy Architecture overview*. Acessado em: agosto de 2024. 2024. URL: https://docs.scrapy.org/en/latest/topics/architecture.html.

[20] Jaime G. Carbonell, Ryszard S. Michalski, and Tom M. Mitchell. "1 - AN OVERVIEW OF MACHINE LEARNING". In: *Machine Learning*. Ed. by Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell. San Francisco (CA): Morgan Kaufmann, 1983, pp. 3–23. ISBN: 978-0-08-051054-5. DOI: https://doi.org/10.1016/B978-0-08-051054-5.50005-4. URL: https://www.sciencedirect.com/science/article/pii/B9780080510545500054.

[21] Inc. Meta Platforms. *Meta Content Library version v5.0*. Acessado em: novembro de 2024. 2024. URL: https://doi.org/10.48680/meta.metacontentlibrary.5.0.

[22] Caroline de Tilia. *X (Twitter) é bloqueado no Brasil; entenda o que vem a seguir*. Acessado em: 09 de dezembro de 2024. 2024. URL: https://forbes.com.br/last/2024/08/x-twitter-e-bloqueado-no-brasil-entenda-o-que-vem-a-seguir/.

[23] Google LLC. *API YouTube Data V3*. Acessado em: novembro de 2024. 2024. URL: https://developers.google.com/youtube/v3/getting-started?hl=pt-br.

[24] Aos fatos. *Site Oficial do Aos Fatos*. Acessado em: dezembro de 2024. URL: https://www.aosfatos.org/noticias/?formato=checagem.

[25] Ni Luh Putu Melyawati et al. "Comparison of Automation Testing On Card Printer Project Using Playwright And Selenium Tools". In: *Journal of Computer Networks, Architecture and High Performance Computing* 6.3 (2024), pp. 1309–1320.

[26] PlayWright. *Playwright for Python*. Acessado em: novembro de 2024. 2024. URL: https://playwright.dev/python/.

APÊNDICES

APÊNDICE A

TABELAS DE METADADOS

link	title	date	author	text	fact	check	tag
https://www.aosfatos.org/i	Vídeo engana ao dizer que desfile	9 de setembro de 2024	Bianca Bortolon	Não é verdade que	Um presidente ser	buscou a transm	falso
https://www.aosfatos.org/i	Perfil de Felipe Neto no Bluesky n	4 de setembro de 2024	Bianca Bortolon	Não é verdade que	Após ajudar promo	Para checar a ve	falso
https://www.aosfatos.org/i	É falso que Anvisa autorizou impo	29 de agosto de 2024	Milena Mangabeira	Não é verdade que	Vacina para varíola	entrou em conta	falso
https://www.aosfatos.org/i	Biden não disse que ele e Kamala	21 de agosto de 2024	Luiz Fernando Menez	Não é verdade que	Biden confessa pla	buscou a gravaç	falso
https://www.aosfatos.org/i	Vídeo de homem ateando fogo a	26 de agosto de 2024	Milena Mangabeira	Não é verdade que	Esses fascistas do	fez uma busca r	nao_e_bem_assim
https://www.aosfatos.org/i	OMS não ordenou que governos p	16 de agosto de 2024	Marco Faustino	Não é verdade que	A OMS ordena que	assistiu à coletiv	falso
https://www.aosfatos.org/i	No Roda Viva, Datena desinforma	13 de agosto de 2024	Bianca Bortolon, Ethel	Em entrevista ao F	"Hospital, por exer	Com a ajuda do I	falso
https://www.aosfatos.org/i	Vídeo mostra show da Madonna i	5 de agosto de 2024	Milena Mangabeira	É falso que um víd	Multidão venezuel	fez uma busca r	falso
https://www.aosfatos.org/i	É falso que ministro da Defesa pr	7 de agosto de 2024	Bianca Bortolon	Não é verdade que	Não é brincadeira.	Por meio de busc	falso
https://www.aosfatos.org/i	Vídeo não mostra venezuelanos c	31 de julho de 2024	Milena Mangabeira	Não é verdade que	Venezuelanos pas	Em uma busca re	falso
https://www.aosfatos.org/i	Gráfico usa dados enganosos par	25 de julho de 2024	Luiz Fernando Menez	Contém dados inc	Como dar certo un	enviou o gráfico	falso
https://www.aosfatos.org/i	Posts usam foto de outra pessoa	18 de julho de 2024	Luiz Fernando Menez	Não é Thomas Ma	Alguém está se pe	fez uma busca r	falso
https://www.aosfatos.org/i	Imagem não prova que Trump foi	15 de julho de 2024	Marco Faustino	Não é verdade que	Trump também foi	comparou a fotc	falso
https://www.aosfatos.org/i	Vídeo mostra muçulmanos celebr	8 de julho de 2024	Milena Mangabeira	Não é verdade que	Comemoração da	Por meio de busc	falso
https://www.aosfatos.org/i	Vídeos que mostram estátuas e fl	26 de junho de 2024	Bianca Bortolon	Não é verdade que			falso
https://www.aosfatos.org/i	Piada de Biden é editada para faz	19 de junho de 2024	Luiz Fernando Menez	São enganosas as	Biden travou outra		falso
https://www.aosfatos.org/i	Vídeo usa legendas falsas para fa	3 de julho de 2024	Bianca Bortolon	Não é verdade que	Acabou o amor. Lu		falso

Fig. 10: Tabela de Metadados extraídos - Fonte: aosfatos.org

link	title	author	date	text	tag
https://www.boatos.org/politica/bolsona	Vídeo que mostra multidão a	Edgard Matsuki	04/09/2024	Análise Uma série de mensag	Fake news ❌
https://www.boatos.org/politica/elon-mu	É falso que Elon Musk tenha	Edgard Matsuki	03/09/2024	Análise O que não tem faltad	Fake news ❌
https://www.boatos.org/esporte/cafu-pr	Golpe de plataforma de inves	Edgard Matsuki	04/09/2024	Análise Uma mensagem viral	Fake news ❌
https://www.boatos.org/tecnologia/notif	Golpe por SMS fala de notific	Edgard Matsuki	04/09/2024	Análise Muitas pessoas recel	Fake news ❌
https://www.boatos.org/esporte/fernanc	É falso que Fernando Diniz te	Edgard Matsuki	05/09/2024	Análise Fernando Diniz é um	Fake news ❌
https://www.boatos.org/tecnologia/lg-fa	Golpe cita quiz da LG por cau	Edgard Matsuki	19/08/2024	Análise Só para não perder o	Golpe ⚠
https://www.boatos.org/saude/siemens-	Nem Siemens lançou medido	Edgard Matsuki	16/08/2024	Análise Recentemente, surgiu	Golpe ⚠
https://www.boatos.org/religiao/padre-m	Golpe cita entrevista falsa do	Edgard Matsuki	21/08/2024	Análise Um vídeo de uma bric	Fake news ❌
https://www.boatos.org/entretenimento/	Mulher que critica Silvio Sant	Edgard Matsuki	22/08/2024	Análise A morte do apresenta	Fake news ❌
https://www.boatos.org/esporte/aeromo	É falso que aeromoça tenha g	Edgard Matsuki	21/08/2024	Análise Uma nova história qu	Fake news ❌
https://www.boatos.org/entretenimento/	Texto que compara "médicos	Edgard Matsuki	null	Análise Quem acompanha o I	Fake news ❌
https://www.boatos.org/saude/medicos-	Texto sobre médicos que mo	Edgard Matsuki	13/08/2024	Análise Pela terceira vez desc	Fake news ❌
https://www.boatos.org/entretenimento/	Homem e menina que cantan	Edgard Matsuki	13/08/2024	Análise Agora em 2024, volto	Fake news ❌
https://www.boatos.org/saude/medicos-	Teoria que fala que médicos	Edgard Matsuki	12/08/2024	Análise A queda do avião na	Fake news ❌
https://www.boatos.org/esporte/atleta-d	Vídeo que está circulando na	Edgard Matsuki	07/08/2024	Análise As Olimpíadas de Par	Fake news ❌
https://www.boatos.org/politica/haddad	É falso que Haddad e Lula ter	Edgard Matsuki	15/08/2024	Análise Já há algum tempo, c	Fake news ❌
https://www.boatos.org/entretenimento/	Vídeo de Henrique Fogaça an	Edgard Matsuki	07/08/2024	Análise Com a chegada do Di	Golpe ⚠

Fig. 11: Tabela de Metadados extraídos - Fonte: boatos.org

link	title	author	date	text	tag
https://g1.globo.com,	É #FAKE que Deolane Bezerra	Por , g1	09/09/2024	Circula nas redes sociais que	#FAKE
https://g1.globo.com,	Veja o que é #FATO ou #FAKE	Por O Globo, g1, TV Glob	06/09/2024	Os jornais O GLOBO e Valor e	#FATO. #NÃOÉBEMASSIM. #FAKE. #FATO.
https://g1.globo.com,	Veja o que é #FATO ou #FAKE	Por , , g1 PE	05/09/2024	O entrevistou ao vivo os três	#NÃOÉBEMASSIM. #FAKE: #FATO: #FATO.
https://g1.globo.com,	Veja o que é #FATO ou #FAKE	Por O Globo, g1, TV Glob	05/09/2024	Os jornais O GLOBO e Valor e	#FATO. #FATO. #FAKE. #FATO. #FATO. #FA
https://g1.globo.com,	É #FAKE que vídeo com show	Por Mel Trindade, TV Gl	04/09/2024	Circula nas redes sociais um	#FAKE
https://g1.globo.com,	É #FAKE que eleitor perde vot	Por Roney Domingos, g1	06/09/2024	Voltou a circular nas redes s	#FAKE
https://g1.globo.com,	Veja o que é #FATO ou #FAKE	Por O Globo, g1, TV Glob	04/09/2024	Os jornais O GLOBO e Valor e	#FATO. #NÃOÉBEMASSIM. #FATO. #FATO.
https://g1.globo.com,	É #FAKE que Alexandre de M	Por Mel Trindade	06/09/2024	Circulam nas redes sociais u	#FAKE
https://g1.globo.com,	Veja o que é #FATO ou #FAKE	Por , , g1 Minas	04/09/2024	Os cinco candidatos à Prefei	#FAKE. #FATO. #NÃOÉBEMASSIM. #FATO.
https://g1.globo.com,	É #FAKE que rótulos apontam	Por Roney Domingos, g1	16/08/2024	Circula nas redes sociais um	#FAKE
https://g1.globo.com,	É #FAKE que piloto de avião c	Por , g1	14/08/2024	Circulam nas redes sociais p	#FAKE
https://g1.globo.com,	É #FAKE que vídeo mostre re	Por , g1 Rio	14/08/2024	Circulou nas redes sociais ur	#FAKE.
https://g1.globo.com,	É #FAKE que médicos mortos	Por g1	15/08/2024	Circula nas redes sociais um	#FAKE
https://g1.globo.com,	É #FAKE que foto de homem	Por Roney Domingos, g1	04/09/2024	Circula nas redes sociais um	#FAKE.
https://g1.globo.com,	É #FAKE que Boulos seja don	Por Mel Trindade, TV Gl	15/08/2024	Circula nas redes sociais um	#FAKE
https://g1.globo.com,	É #FAKE que vídeo mostre pa	Por , g1	14/08/2024	Circula nas redes sociais um	#FAKE
https://g1.globo.com,	É #FAKE que filha de Silvio Sa	Por g1	19/08/2024	Circulam em serviços de me	#FAKE

Fig. 12: Tabela de Metadados extraídos - Fonte: g1.globo.com

link	title	date	author	text	tag	subject
https://lupa.uol.com	Em Porto Alegre, Ma	02.09.2024 - 19h30	Carol Macário ,	Maria do Rosário (P	Falso - Falta contexto - Verda	Eleições 2024
https://lupa.uol.com	Camozzato erra valo	04.09.2024 - 16h41	Ítalo Rômany , M	O deputado estadua	Falso - Subestimado - Exager	Eleições 2024
https://lupa.uol.com	No Roda Viva, Nunes	10.09.2024 - 19h50	Catiane Pereira	O prefeito Ricardo N	Falso - Verdadeiro - Falta con	Eleições 2024
https://lupa.uol.com	Em Porto Alegre, Me	03.09.2024 - 18h30	Carol Macário ,	O prefeito de Porto A	Falso - Verdadeiro - Exagerad	Eleições 2024
https://lupa.uol.com	Principais candidato	09.09.2024 - 15h27	Maiquel Rosaur	A enchente que dest	Falta contexto - Falso - Falta c	Porto Alegre
https://lupa.uol.com	Em Porto Alegre, Jul	10.09.2024 - 15h00	Carol Macário ,	A ex-deputada estad	Exagerado - Verdadeiro - Verc	Eleições 2024
https://lupa.uol.com	No Roda Viva, Março	03.09.2024 - 19h30	Catiane Pereira	O empresário Pablo	Falso - Exagerado - Exagerad	Eleições 2024
https://lupa.uol.com	No Roda Viva, Daten	13.08.2024 - 19h30	Carol Macário ,	O apresentador José	Exagerado - Verdadeiro - Verc	Eleições 2024
https://lupa.uol.com	Na Globonews, Tarcí	27.08.2024 - 17h30	Catiane Pereira	O deputado federal T	Exagerado - Verdadeiro - Verc	Rio
https://lupa.uol.com	Candidatos do Rio er	09.08.2024 - 20h35	Carol Macário ,	O primeiro debate er	Verdadeiro - Verdadeiro - Verc	Eleições 2024
https://lupa.uol.com	Na GloboNews, Ram	28.08.2024 - 20h00	Gabriela Soares	O deputado federal A	Falso - Falso - Verdadeiro - Fa	Eleições 2024
https://lupa.uol.com	No Roda Viva, Boulo	27.08.2024 - 19h30	Carol Macário ,	O deputado federal C	Exagerado - Verdadeiro - Verc	São Paulo
https://lupa.uol.com	Na Globonews, Paes	29.08.2024 - 20h59	Evelyn Fagunde	O prefeito Eduardo P	Falso - Verdadeiro - Subestim	Eleições 2024
https://lupa.uol.com	Tabata erra dados sc	20.08.2024 - 18h00	Catiane Pereira	A deputada federal T	Verdadeiro - Verdadeiro - Exa	Capital paulista
https://lupa.uol.com	Ministra da Saúde er	19.04.2024 - 14h30	Carol Macário	Em audiência na últi	Falso - Verdadeiro - Verdadeir	Política
https://lupa.uol.com	Ato de 1º de maio: Lu	01.05.2024 - 19h00	Ítalo Rômany	O presidente Luiz Ina	Exagerado - Verdadeiro - Fals	Checagem
https://lupa.uol.com	Ministro Silvio Costa	20.03.2024 - 17h44	Carol Macário	Em entrevista ao , o	Falso - Verdadeiro - Falta con	Roda viva

Fig. 13: Tabela de Metadados extraídos - Fonte: lupa.com.br

link	title	author	date	text
https://noticias.uol.com	Comprova lança pacote de recursos para apoiar c	Projeto Comprova	05/09/2024 14h42	A partir desta quinta-feira, 5 de se
https://noticias.uol.com	Em sabatina, Marçal erra sobre Craco Resiste e di	Isabela Aleixo	04/09/2024 15h36	Em , o candidato à Prefeitura de S
https://noticias.uol.com	É falso que pessoas vacinadas tenham o dobro d	Projeto Comprova	03/09/2024 15h29	Conteúdo investigado : Autor de
https://noticias.uol.com	É falso que osso de patinho passou a ser vendido	Projeto Comprova	04/09/2024 13h14	Conteúdo investigado : Em , hom
https://noticias.uol.com	É falso vídeo em que Pablo Marçal supostamente	Isabela Aleixo	04/09/2024 14h09	É falso que Pablo Marçal tenha di
https://noticias.uol.com	Não é possível acessar internet da Starlink sem ki	Isabela Aleixo	03/09/2024 16h55	É falso que usuários possam se c
https://noticias.uol.com	Vídeo em que deputada dos EUA critica Moraes é	Thâmara Kaoru	04/09/2024 17h24	Um vídeo em que uma deputada e
https://noticias.uol.com	Mulher que critica Alexandre de Moraes em vídeo	Ricardo Espina	04/09/2024 17h56	Uma mulher que aparece em um v
https://noticias.uol.com	Lei sobre terras incendiadas não acabou com que	Thâmara Kaoru	05/09/2024 14h17	É falso que uma lei da Espanha su
https://noticias.uol.com	É falso que interior de avião tenha sido tomado p	Ricardo Espina	04/09/2024 14h47	O vídeo que mostra um avião sup
https://noticias.uol.com	Vídeo de Lula e Moraes se beijando é falso e foi c	Ricardo Espina	03/09/2024 17h06	É falso o vídeo que circula pelas r
https://noticias.uol.com	É falso que homem ao lado de Bolsonaro seja o m	Ricardo Espina	03/09/2024 15h11	É falso que um homem que apare
https://noticias.uol.com	Pablo Marçal: desinformação como estratégia co	Thaís Lazzeri*	30/08/2024 05h30	"Quem não quer correr risco fica e
https://noticias.uol.com	Não há evidências de que MST tenha ameaçado 'i	Projeto Comprova	30/08/2024 16h43	Conteúdo investigado : atribuída
https://noticias.uol.com	Vídeo que mostra suposto roubo em fazenda é de	Projeto Comprova	30/08/2024 16h08	Conteúdo investigado : exibem v
https://noticias.uol.com	Caso Marçal x Boulos: saiba como fazer uma bus	Projeto Comprova	30/08/2024 17h59	Conteúdo analisado : A Folha pub
https://noticias.uol.com	Supla não se referia a Lula ao dizer 'se roubar con	Ricardo Espina	02/09/2024 16h46	O cantor Supla não se referia a Lu

Fig. 14: Tabela de Metadados extraídos - Fonte: noticias.uol.com.br

APÊNDICE B
CÓDIGOS DESENVOLVIDOS

```python
import scrapy

class AosfatosSpider(scrapy.Spider):
    name = "aosfatos"
    #allowed_domains = ["aosfatos.com"]
    start_urls = ["https://www.aosfatos.org/noticias/?formato=checagem"]

    count = 0
    max_count = 400

    def parse(self, response):
        if self.count >= self.max_count:
            return

        for manchete in response.css('.grid'):
            link = manchete.css('div a::attr(href)').get()

            if self.count < self.max_count:
                self.count += 1

            yield response.follow(link, self.parse_article)

        next_page = response.css('.text-center a::attr(href)').getall()[-1]
        if next_page is not None:
            yield response.follow(next_page, self.parse)

    def parse_article(self, response):
        dados = {
            'link': response.url,
            'title': response.css('.prose h1::text').get(),
            'data': response.css('.prose aside::text').get(),
            'author': response.css('.prose aside::text').getall()[6],
            'text': ' '.join(response.css('.mb-11 p::text').getall()).replace('\r\n', ''),
            'fact': response.css('.prose blockquote p::text').get() or response.css('.prose
                blockquote::text').get(),
            'check': ' '.join(response.css('.mb-11 details p::text').getall()),
            'tag': response.css('.prose blockquote::attr(data-stamp)').get()
        }
        yield dados
```

Código 1: Código em python do spider para o site aosfatos.org

```python
def search_youtube(q, region_code='BR', video_duration='any', video_definition='any',
                   max_results=50, page_token=None, published_after=None, published_before=None
                       ):
    request = youtube.search().list(
        part="id,snippet",
        type='video',
        q=q,
        regionCode=region_code,
        videoDuration=video_duration,
        videoDefinition=video_definition,
        maxResults=max_results,
        publishedAfter=published_after,
        publishedBefore=published_before,
        fields="kind, etag, nextPageToken, regionCode, pageInfo, items(id(videoId),snippet(
            publishedAt,channelId,channelTitle,title,description))",
        pageToken=page_token
    )
    search_response = request.execute()

    for item in search_response['items']:
        video_id = item['id']['videoId']
        stats = get_video_statistics(video_id)
```

```
        item['statistics'] = stats['items'][0]['statistics']

    return search_response

def get_video_statistics(video_id):
    request = youtube.videos().list(
        part="statistics",
        id=video_id
    )
    return request.execute()

def extract_comment_metadata(comment_thread):
    comment = comment_thread['snippet']['topLevelComment']['snippet']
    return {
        'authorDisplayName': comment.get('authorDisplayName'),
        'textDisplay': comment.get('textDisplay'),
        'likeCount': comment.get('likeCount')
    }

def get_video_comments(video_id, max_results=100):
    try:
        request = youtube.commentThreads().list(
            part="snippet",
            videoId=video_id,
            maxResults=max_results,
            textFormat="plainText"
        )
        response = request.execute()

        comments_metadata = []
        for comment_thread in response['items']:
            comment_data = extract_comment_metadata(comment_thread)
            comments_metadata.append(comment_data)

        return comments_metadata
    except HttpError as e:
        if e.resp.status == 403 and 'commentsDisabled' in str(e):
            print(f"Coment rios desativados para o v deo {video_id}")
            return None
```

Código 2: Script em python para utilização Youtube Data v3 API

```python
def get_video_transcript(video_id):
    try:
        transcript = YouTubeTranscriptApi.get_transcript(video_id, languages=['pt'])

        full_transcript = " ".join([t['text'] for t in transcript])
        return full_transcript
    except TranscriptsDisabled:
        return "Transcri  o  n  o  dispon vel"
    except NoTranscriptFound:
        return "Nenhuma  transcri  o encontrada"
    except Exception as e:
        return f"Erro ao obter  transcri  o: {e}"
```

Código 3: Script em python para utilização Youtube Transcripts API

```python
def download_audio(video_ids, save_path='./downloads'):
    os.makedirs(save_path, exist_ok=True)

    ydl_opts = {
        'format': 'bestaudio/best',
        'postprocessors': [{
            'key': 'FFmpegExtractAudio',
            'preferredcodec': 'mp3',
            'preferredquality': '192',
        }],
        'outtmpl': f'{save_path}/%(title)s.%(ext)s',
        'ffmpeg_location': r'C:\Users\thorf\Documents\ffmpeg\bin',
    }

    with yt_dlp.YoutubeDL(ydl_opts) as ydl:
        for video_id in video_ids:
            try:
                url = f"https://www.youtube.com/watch?v={video_id}"
                ydl.download([url])
                print(f" udio  do  v deo  {video_id} baixado com sucesso.")
            except Exception as e:
                print(f"Erro ao baixar o  udio  do  v deo  {video_id}: {e}")
```

Código 4: Script em python para utilização yt-dlp API

APÊNDICE C
RESULTADOS DAS ANÁLISES DE SENTIMENTO

videoId	1º Comentário	1º Sentimento	2º Comentário	2º Sentimento	3º Comentário	3º Sentimento
5cPfkm-BvPg	Sandro Mabel	positivo	Aliança ao Gustavo Gayer Fr Ao menos um vice do NOVO	muito negativo	55:10 que candidato caricato	muito positivo
5AybpQ7-G3s	2 2 BR 2 2 BR 2	muito positivo	Concordo plenamente,, Deus	muito positivo		
XL-GcVmKdNQ	Ela tem soluções p	positivo				
CTf5h02KqTk	Candidata esquerd	muito negativo	É pt não presta. Goiânia não	muito negativo	Prefeita em 2025.	positivo
-pGU4Jht9VM	Fred 2 2	muito positivo	Que entrevista péssima, perd Pra que serviu essa "sabatina	muito negativo	Moro em Senador Canedo e	muito negativo
zOPFfZhreTI	MMMMMMMMM	muito negativo	FAZ O M M M	muito negativo	parece fake	muito negativo
xwtmJZ35o8o	Cade de Goiânia???	muito negativo				
OE4RyYWP2cM	Fred 2 2 BR	muito positivo	Mentalidade rodoviária .. Mal Qro ver a proposta que tem p	negativo	Goiânia terá um futuro próspe	muito positivo
9Ct-AjkUWmY	Sou candidato a ve	muito negativo	Esse cara e a melhor opção p	muito positivo	Sou Washington Luiz candida	muito positivo
ETYhH86xYrg		muito positivo	Bom dia Caros Jornalistas Ar	muito positivo	M, Pablo Marçal...	neutro
aD_DM34c3BA	Kkkkkkkkkkkkkkkkkl	muito negativo	Se o Lula não gosta de Pablo	neutro	Olha o que a raposa disse ,da	muito negativo
vFGulaiDR10	Nenhum serra tão i	muito positivo	agora é Vanderlan!	muito positivo	Vanderlan votou no Dino e no	muito negativo
qtRvEc_y8vs	2 8 BRBRBRBRBRBR	muito positivo	E as coisas foram evoluindo	muito positivo		muito positivo
aocWT2IhoAY	Famoso, playboy in	muito positivo	Olha acho que não é fácil Ma	muito negativo	Cadeira é pra sentar mas o D	muito negativo
Mo4rEB_c0BU	Lugaf dele é na cac	muito positivo	Uai...até onde eu sei,era o Dó	muito negativo	Uol lixo. Só sabe atacar Pablo	muito negativo
l9f_rz0wnA8	"população, é um fa	muito negativo		muito positivo	Romu... Eu pensei que era "ra	muito negativo
geN7NKIdahA	#Sandro2024 #Mal	neutro	Tem boas propostas mas per	negativo	Arrocha Goiânia	muito negativo

Fig. 15: Tabela de comentários dos vídeos com seus sentimentos

videoId	transcription	sentiment
lW8W20c20Wk	definiram Goiânia uma surpresa Fred Rodrigues o Yuri Pita falava ontem tem que dar os créditos ele apo	muito negativo
dDBnfpqk2O4	ASSIM, COM ESSA RESENHA DE VOCÊS MAS, VAMOS LÁ DE NOVO COM A NOSSA COBERTURA DAS E	negativo
UrE8EP8RVcM	[Música] Breaking News CNN tá aqui na capital Goiana Goiânia Sandro Mabel da União Brasil é o novo pr	muito negativo
G2FGTKF9iWw	DA CAMPOS CONTA AQUI EM GOIÂNIA OS CANDIDATOS FREDDY RODRIGUES DO PL SANDRO MABEL	muito negativo
lKPbB_WDG_0	vamos vamos até dar uma olhada também em outro resultado acaba de sair ã um cenário também que	muito negativo
9I2I6impn3I	definição já em Goiânia capital de Goiás vamos ver os números lá estão chegando ainda a lá e Sandro M	muito negativo
XkgjQVi066M	falando ainda do cenário em Goiás porque por lá a gente tem uma divisão da direita Ju comentávamos s	muito positivo
QZLYvYTeF0I	Mabel está eleito tá aí matematicamente matematicamente não tem mais virada então a Prefeitura de G	muito negativo
AWG6lHf--5M	vamos dar uma olhada em Porto Alegre parece que tem resultado também alos e segundo turno já saiu	muito negativo
7xwV9Bcv_wg	Marina pelo amor de Deus as diretoras deviam amar a Marina sabe as professoras ai minha querida A M	neutro
MId5I-D0Icg	uma capital que ainda não discutimos que é Goiânia havia ali uma possibilidade de PTI para o segundo t	muito negativo
mrreETCQ6C0	primeiro por que eu estou inelegível me reuni com traficantes no morro do Alemão por acaso não né me	muito negativo
_jsfk3-4ItI	a gente tem eleição matematicamente definida 94,2 das urnas apuradas ou seja os 58% de urnas a serer	muito negativo
OUq08r05QqM	pesquisa Quest indica possibilidade de virada no segundo turno em duas de 15 capitais que ainda não d	muito negativo
5AybpQ7-G3s	eleições 2024 é aqui na PUC TV nós estamos dando continuidade estamos entrevistando todos os candi	muito positivo
d55nyWYnjSg	eleições 2024 é aqui na PUC TV e hoje nós damos continuidade às entrevistas com os candidatos que e	muito positivo
dSsPPSQrvKI	resultado definido em Goiânia foi para segundo turno também tem segundo turno ah em Fortaleza a gen	muito negativo

Fig. 16: Tabela de transcrições dos vídeos com seus sentimentos