UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO MULTIDISCIPLINAR

IVO SANTOS PAIVA
FLÁVIO ALVES WILDNER REAL ROSA

Uma ferramenta para coleta de informações no Instagram

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

Nova Iguaçu, Julho de 2022

Uma ferramenta para coleta de informações no Instagram

Ivo Santos Paiva

Flávio Alves Wildner Real Rosa

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

Apresentado por:

_____

Ivo Santos Paiva

_____

Flávio Alves Wildner Real Rosa

Aprovado por:

_____

Prof. Filipe Braida do Carmo, D.Sc.

_____

Prof. Leandro Guimarães Marques Alvim, D.Sc.

_____

Prof. Bruno José Dembogurski, D.Sc.

NOVA IGUAÇU, RJ - BRASIL

Julho de 2022

Emitido em 13/07/2022

DOCUMENTOS COMPROBATÓRIOS Nº 12958/2022 - CoordCGCC (12.28.01.00.00.98)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 18/07/2022 12:12)
BRUNO JOSE DEMBOGURSKI
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: 2124964

(Assinado digitalmente em 14/07/2022 10:01)
FILIPE BRAIDA DO CARMO
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: 3929524

(Assinado digitalmente em 15/07/2022 16:04)
LEANDRO GUIMARAES MARQUES ALVIM
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: 1800852

(Assinado digitalmente em 13/07/2022 16:08)
FLÁVIO ALVES WILDNER REAL ROSA
DISCENTE
Matrícula: 2014780194

(Assinado digitalmente em 13/07/2022 18:22)
IVO SANTOS PAIVA
DISCENTE
Matrícula: 2014780291

Para verificar a autenticidade deste documento entre em https://sipac.ufrrj.br/documentos/ informando seu número: **12958**, ano: **2022**, tipo: **DOCUMENTOS COMPROBATÓRIOS**, data de emissão: **13/07/2022** e o código de verificação: **1409a0eefe**

Agradecimentos

Ivo Santos Paiva

Primeiramente, gostaria de agradecer aos meus pais: Ricardo e Mariangela, que estiveram presentes em todas as etapas da minha vida, vibrando com cada conquista, me motivando e cobrando sempre que necessário. Vocês são o meu porto seguro, e devo muito a vocês por tudo que me ensinaram e me proporcionaram. Só cheguei até aqui graças ao valor que vocês sempre deram à minha educação e a do meu irmão.

Agradeço também ao meu irmão, Hugo, que sempre esteve ao meu lado, me motivando e inspirando a ser uma pessoa melhor.

Ao Flávio, minha dupla neste trabalho, que esteve comigo durante todo o desenvolvimento deste projeto, pela parceria e ajuda ao longo dessa jornada;

A todos os professores do Departamento de Ciência da Computação da Universidade Federal Rural do Rio de Janeiro, pela dedicação que puseram em suas aulas e pela qualidade do ensino prestado. Em especial, ao meu orientador Filipe Braida pela paciência que teve comigo (mesmo "enrolando" o TCC por tanto tempo), e pela enorme dedicação com que sempre ministrou os seus cursos;

Aos amigos que fiz na Rural, que tornaram a experiência muito mais leve e agradável, em especial Gustavo e Rodrigo, meus parceiros dos trabalhos em grupo e ex-colegas de estágio, amigos para os momentos de descontração e para os "perrengues";

Muito obrigado!

Flávio Alves Wildner Real Rosa

Quero agradecer aos meus pais, Ivandréa e José, que não tiveram a oportunidade de acesso aos estudos, mas desde sempre me incentivaram e me deram todo apoio para focar na educação.

Agradeço também ao meu irmão, Josivan, que esteve sempre junto comigo em todos os momentos de dificuldade, principalmente nos problemas familiares, para seguir firme durante o curso e minha carreira profissional. Além também de sua sensibilidade, que em nossas conversas da madrugada, me fizeram evoluir muito como pessoa.

A minha dupla deste trabalho, Ivo, que se manteve junto comigo por todo esse tempo e pela enorme dedicação que teve para fazer esse trabalho acontecer.

Ao excelente corpo docente do curso de Ciência da Computação na Universidade Federal Rural do Rio de Janeiro, que proporcionaram aulas de altíssima qualidade e de forma leve.

E também, aos amigos que fiz durante o curso, que fizeram toda a diferença com as parcerias nos trabalhos práticos e nas sessões de estudos para as temidas provas.

Ademais, agradeço a todos que conviveram comigo durante esse processo, que, de alguma forma, contribuíram para minha formação.

RESUMO

Uma ferramenta para coleta de informações no Instagram

Ivo Santos Paiva e Flávio Alves Wildner Real Rosa

Julho/2022

Orientador: Filipe Braida do Carmo, D.Sc.

O Instagram é uma das principais redes sociais utilizadas globalmente, e uma das maiores em termos de usuários mensais. Com sua enorme base de usuários e interações realizadas, ela tem chamado a atenção de pesquisadores para realização de análises de comportamento coletivo, além de detecção de traços de depressão a partir de análises de postagens. Com o crescimento da demanda por estudos que contemplem o Instagram, técnicas eficientes para a extração de informação a partir da rede social se mostram de suma importância. Nesse contexto, o presente trabalho busca apresentar uma forma independente e modular de coleta de informações, que possa ser alterada de acordo com o interesse do usuário e que possibilite o escalonamento de funções específicas de domínio. Na abordagem apresentada, o usuário poderá instanciar apenas os módulos que lidem com a informação específica de trabalho, além de possibilitar o desenvolvimento de módulos adicionais, contemplando futuras funcionalidades implementadas pela rede social. Com o trabalho apresentado, espera-se ampliar a gama de ferramentas disponíveis para a coleta de informações no Instagram e, dessa forma, facilitar o desenvolvimento de novas pesquisas englobando a rede social.

ABSTRACT

Uma ferramenta para coleta de informações no Instagram

Ivo Santos Paiva and Flávio Alves Wildner Real Rosa

Julho/2022

Advisor: Filipe Braida do Carmo, D.Sc.

Instagram is one of the main social networks used globally, and one of the largest in terms of monthly users. With its huge user base and interactions carried out, it has drawn the attention of researchers to perform collective behavior analysis, in addition to detecting depression traits from post analysis. With the growing demand for studies that include Instagram, efficient techniques for extracting information from the social network are of huge importance. In this context, the present work seeks to present an independent and modular way of collecting information, which can be changed according to the user's interest and which allows the scaling of specific domain functions. In the presented approach , the user will be able to instantiate only the modules that deal with the specific information of interest, in addition to enabling the development of additional modules, contemplating future functionalities implemented by the social network. With the work presented, it is expected to expand the range of tools available for collecting information on Instagram and, in this way, facilitate the development of new research that include the social network.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

API	*Application Programming Interface*
CSV	*Comma Separated Values*
DDoS	*Distributed Denial of Service*
DIKW	*Data Information Knowledge Wisdom*
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
JSON	*JavaScript Object Notation*
KDD	*Knowledge Discovery in Databases*
ORM	*Object-Relational Mapping*
SGBD	*Sistema Gerenciador de Banco de Dados*
SQL	*Structured Query Language*
URL	*Uniform Resource Locator*

Sumário

Capítulo 1

Introdução

Atualmente, vivemos em uma era de constante atualização. O que começou na década de 1980 com a criação da Internet hoje já apresenta um cenário em que um universo completamente virtual, onde é possível trabalhar em um escritório com mesa e cadeira que só existem virtualmente, já começa a virar realidade. Nesse novo universo, milhares de informações são geradas a cada minuto e com enorme facilidade deixam de ser novidade, se tornando completamente desatualizadas e sem importância.

Muito desse cenário se deve ao surgimento das redes sociais, onde milhões de pessoas compartilham diariamente recortes de suas experiências, montando constantemente a história da civilização do século XXI.

Compartilhamento de fotos, vídeos e mensagens são apenas algumas das possibilidades trazidas pelas redes sociais, que visam, cada vez mais, manter seus usuários mais conectados, possibilitando a realização da maior quantidade possível de atividades dentro de suas plataformas. Toda essa exposição aos ambientes virtuais tem trazido grandes mudanças no comportamento e na vida de quem se encontra ativamente incluído no mundo virtual, e entender esses movimentos é de extrema importância.

Reece e Danforth (2017) realizaram um estudo utilizando dados do Instagram[1]

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

para gerar um modelo capaz de identificar marcadores de depressão a partir do comportamento na rede social. Utilizando análise de cores, metadados do Instagram e algoritmos de detecção de face, os autores conseguiram identificar que fotos postadas por indivíduos com depressão apresentavam maior inclinação para as cores azul e cinza, além de demonstrar uma preferência pela utilização de filtros monocromáticos.

Em outro estudo, Aragão et al. (2016) analisam o Instagram como ambiente de marketing e sua influência no processo de compra de um consumidor, funcionando como ferramenta para construção da reputação de empresas e um termômetro para as diferentes abordagens. Entre os resultados encontrados, os autores concluíram que os comentários geram maior impacto no fator vendas, quando comparado às interações com o botão "curtir", influenciando os antecedentes da decisão de compra como nível de confiança e risco percebido.

Esses são apenas alguns dos trabalhos que tratam da rede social, enfatizando a sua importância para o meio acadêmico. Para que os trabalhos sejam viabilizados, é necessária uma base significativa de dados de forma a servir de embasamento para a tomada de conclusões e utilização em modelos preditivos ou tratamento de imagens. O acesso a esses dados, no entanto, não é facilitado, uma vez que — devido às políticas de privacidade — não é fornecida uma API oficial que contemple um conjunto amplo de informações tal como realizado por outras redes sociais, a exemplo do Twitter[2].

Com o intuito de fornecer uma alternativa moldável e escalável para a captura de dados e que não seja dependente de permissões prévias ou de manutenção de APIs por parte da plataforma, este trabalho propõe a construção de um *web scraper* modularizado e escalável para o Instagram. Espera-se que com esse trabalho, novos estudos possam ser viabilizados e que novos módulos sejam acoplados, à medida que haja a necessidade de captura de novas informações ou que novas seções sejam implementadas pela rede social.

O trabalho está dividido em cinco capítulos, a contar desta introdução. No segundo capítulo é apresentado o conceito de *web scraper*, além do processo de descoberta

[2]<https://twitter.com/>

de conhecimento. No capítulo seguinte, são apresentados: a motivação do trabalho, os trabalhos relacionados e uma descrição mais detalhada da proposta de trabalho. No capítulo quatro estão dispostas as decisões realizadas para a implementação do projeto, junto com uma breve descrição das tecnologias utilizadas, e uma apresentação do projeto implementado. O último capítulo apresenta as conclusões do trabalho e as propostas de trabalhos futuros.

Capítulo 2

Fundamentação

Nesse capítulo, é tratado o campo da Descoberta de Conhecimento, apresentando o processo e suas etapas, junto com a Hierarquia do Conhecimento, além de apresentar uma descrição para a técnica de *Web Scraping*.

2.1 Descoberta de Conhecimento

Sistemas da informação existem há milhares de anos, muito antes do surgimento das tecnologias de informação e comunicação atuais (BASKARADA; KORONIOS, 2013). Baskarada e Koronios (2013) ressaltam que a hierarquia Dados, Informação, Conhecimento e Sabedoria (do inglês DIKW, *Data, Information, Knowledge, Wisdom*), também referenciada como hierarquia de dados, hierarquia da informação e hierarquia do conhecimento, forma a base das pesquisas de sistemas da informação, sendo usada amplamente na literatura, ainda que não exista um consenso entre as definições dos elementos que a compõem.

Rowley (2007) faz um estudo das diferentes definições dadas para os elementos da hierarquia, citando Ackoff (1989) como uma das primeiras e principais referências dessa estrutura, e observa que, tipicamente, cada nível da hierarquia é definido em termos do nível imediatamente anterior, tendo dados como nível base e sabedoria como nível mais alto. Na tabela 2.1, é possível identificar algumas definições para os

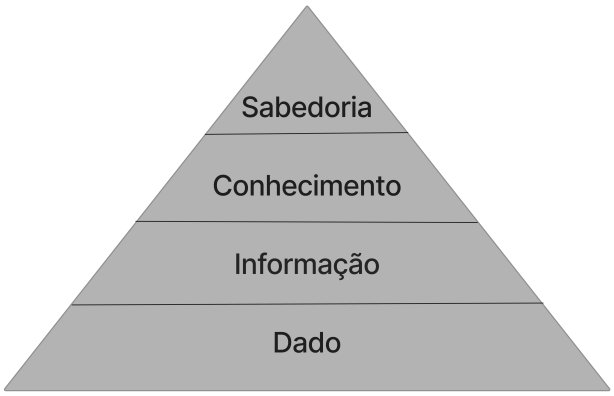

Figura 2.1: Hierarquia DIKW adaptada de Rowley (2007)

elementos da hierarquia.

A transição entre os níveis da hierarquia DIKW envolve adição de contexto e processamento por parte de quem está realizando as análises. Inicialmente, a utilização da informação para a geração de conhecimento que possa ser utilizado para a tomada de decisões e resolução de problemas era realizada de forma manual, sendo feita por especialistas que analisavam as informações obtidas e geravam relatórios que seriam posteriormente utilizados para a definição das estratégias de negócio (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Ainda segundo Fayyad, Piatetsky-Shapiro e Smyth (1996), para muitas aplicações, essa forma manual de tratamento de dados é lenta, cara e altamente subjetiva, se tornando completamente impraticável com o crescimento acelerado da disposição de dados.

Com o intuito de tornar possível o processamento da grande quantidade de dados e informações geradas por meio dos avanços tecnológicos, surgiu — da interseção de três áreas: estatística clássica, inteligência artificial e aprendizado de máquina — a tecnologia de Mineração de Dados (*Data Mining*) (CORRÊA; SFERRA, 2003). Essa tecnologia apresenta grande interseção com o processo de KDD (*Knowledge Discovery in Databases*) — Descoberta de Conhecimento em Bases de Dados, em português —, sendo tratada como sinônimos por alguns autores e como uma etapa do KDD por outros (CAMILO; SILVA, 2009).

Fayyad, Piatetsky-Shapiro e Smyth (1996) ressaltam que o termo KDD foi

Item	Descrição
Dados	Os dados não têm significado ou valor porque estão sem contexto e interpretação (JESSUP; VALACICH, 2003; BOCJI et al., 2003; GROFF; JONES, 2003)
	Dados são fatos ou observações discretas e objetivas, desorganizadas e não processadas e que não transmitem nenhum significado específico (AWAD; GHAZIRI, 2004; CHAFFEY; WOOD, 2005; PEARLSON; SAUNDERS, 2004)
	Dados são uma descrição elementar e registrada de coisas, eventos, atividades e transações (LAUDON; LAUDON, 2004; TURBAN; RAINER; POTTER, 2005; BODDY; BOONSTRA; KENNEDY, 2005)
Informações	Informações são dados processados para um propósito
	A informação é um dado formatado [...(e)] pode ser definido como uma representação da realidade (JESSUP; VALACICH, 2003, p. 7)
	Informação são dados que foram organizados para que tenham significado e valor para o destinatário (TURBAN; RAINER; POTTER, 2005; BODDY; BOONSTRA; KENNEDY, 2005)
Conhecimento	Conhecimento são dados e/ou informações que foram organizadas e processadas para transmitir compreensão, experiência, aprendizado acumulado e conhecimento à medida que se aplicam a um problema ou atividade atual (TURBAN; RAINER; POTTER, 2005, p. 38)
	Conhecimento é a combinação de dados e informações, aos quais se adiciona a opinião de especialistas, habilidades e experiência, para resultar em um ativo valioso que pode ser usado para auxiliar na tomada de decisões (CHAFFEY; WOOD, 2005, p. 223)
Sabedoria	Conhecimento acumulado, que permite entender como aplicar conceitos de um domínio a novas situações ou problemas (JESSUP; VALACICH, 2003)

Tabela 2.1: Definições de Dados, Informações, Conhecimento e Sabedoria, a partir dos estudos realizados por (ROWLEY, 2007)

cunhado em 1989 para enfatizar que o conhecimento é o produto final de uma descoberta orientada a dados, adicionando que "*Data Mining* é a aplicação de

algoritmos específicos para extração de padrões a partir de dados", sendo apenas uma etapa do KDD, definindo este como "o processo não trivial de identificar padrões válidos, novos, potencialmente úteis e, em última análise, compreensíveis em dados", e destacam:

> As etapas adicionais no processo KDD, como preparação de dados, seleção de dados, limpeza de dados, incorporação de conhecimentos prévios apropriados, e interpretação adequada dos resultados de mineração, são essenciais para garantir que seja obtido conhecimento útil. A aplicação cega de métodos de mineração de dados (...) pode ser uma atividade perigosa, levando facilmente à descoberta de dados sem sentido e padrões inválidos. (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996, tradução nossa)

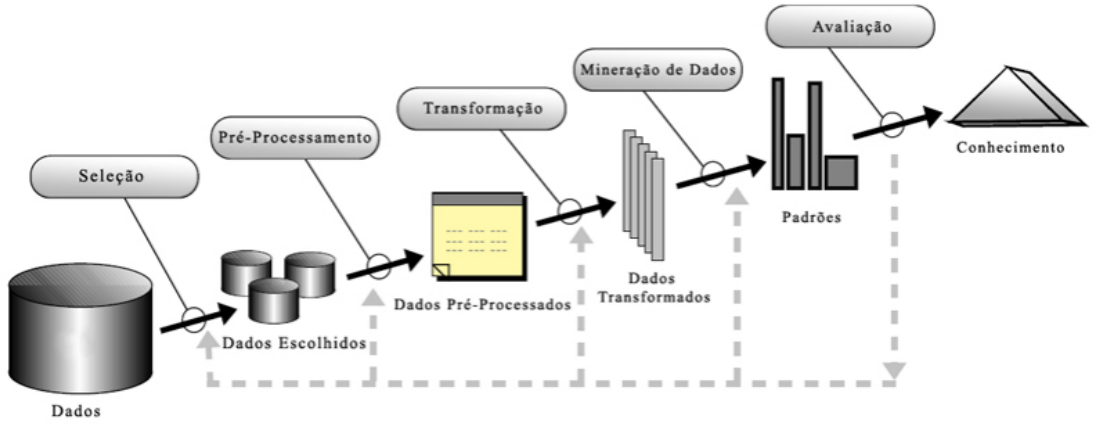

Figura 2.2: Visão Geral do Processo de KDD, retirado de (CAMILO; SILVA, 2009) em adaptação a (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996)

Embora haja divergências entre autores com relação a sua divisão e classificação, o processo de KDD pode ser dividido em 5 etapas: Seleção, Pré-Processamento, Transformação, Mineração de Dados e Avaliação, conforme mostrado na figura 2.2, sendo um processo interativo e iterativo, envolvendo várias etapas com muitas decisões tomadas pelo usuário (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Para GOLDSCHMIDT e Passos (2005), as etapas de KDD podem ser agrupadas em 3 grupos operacionais: Pré-Processamento, Mineração de Dados e Pós-

Processamento. No primeiro são realizadas 4 funções principais:

- **Seleção de Dados**, onde ocorre a identificação das informações que serão efetivamente consideradas ao longo do processo, restringindo os atributos e os registros relevantes;
- **Limpeza dos Dados**, que contempla os tratamentos realizados para garantir a qualidade, eliminando qualquer inconsistência e corrigindo informações errôneas ou incompletas que representem ruídos ou anomalias no conjunto;
- **Codificação dos Dados**, onde estes são tratados para que fiquem em um formato que poderá ser utilizado na etapa de mineração, podendo ser representado por categorias ou intervalos;
- **Enriquecimento dos Dados**, onde, na medida do possível, os dados são complementados com informações que possam tornar a descoberta de conhecimento mais eficaz, podendo fazer uso de bases externas de dados para tal.

Na sequência, durante a etapa de Mineração de Dados, é realizada a descoberta efetiva do conhecimento, sendo a etapa mais importante do processo de KDD (GOLDSCHMIDT; PASSOS, 2005). Nessa etapa, deve ser feita a escolha das técnicas e modelos que mais se adequem ao conjunto de dados trabalhado, uma vez que cada tipo de problema pode exigir uma abordagem específica (CORRÊA; SFERRA, 2003).

Finalizando o processo de KDD, na etapa de Pós-Processamento, é realizada a interpretação e o tratamento do conhecimento obtido na Mineração de Dados, podendo incluir simplificações com elaboração de gráficos e diagramas para melhor representação dos resultados obtidos, além de verificações para determinar a necessidade de retornar a alguma etapa do processo (GOLDSCHMIDT; PASSOS, 2005; CORRÊA; SFERRA, 2003).

2.1.1 Mineração de Dados

A Mineração de Dados apresenta dois objetivos principais: predição e descrição. Na predição são realizadas inferências a partir de variáveis, ou campos da base de

dados, para prever valores futuros ou desconhecidos de outras variáveis de interesse. Na descrição, o foco se dá na descoberta de padrões que descrevam as propriedades gerais dos dados, e que possam ser interpretados por humanos (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

A classificação entre os objetivos na Mineração de Dados se dá de acordo com as tarefas executadas e as diferentes abordagens existentes. Dentre elas, destacam-se:

- **Classificação** - Visa identificar a classe a qual um registro pertence. Nela, um conjunto de registros já classificados é analisado a fim identificar uma função ou modelo que seja capaz de mapear os registros em rótulos ou classes pré-definidas. Uma vez encontrada, essa função é aplicada a novos registros a fim de descobrir em que categoria estes se encontram (GOLDSCHMIDT; PASSOS, 2005; CAMILO; SILVA, 2009).

- **Agrupamento/Clusterização** - Separa os registros em subconjuntos ou *clusters* a partir de características em comum, de forma que elementos dentro de um *cluster* apresentem alta similaridade ao passo que a semelhança entre *clusters* distintos seja mínima. Ao contrário da Classificação, no agrupamento as classes ainda não estão definidas, podendo, inclusive, ser utilizado para gerar os rótulos de classe (GOLDSCHMIDT; PASSOS, 2005; CORRÊA; SFERRA, 2003).

- **Regressão** - De maneira similar à tarefa de Classificação, realiza inferências a partir de registros já classificados/parametrizados, diferenciando-se por realizar o mapeamento para valores reais ao invés de rótulos categóricos (CAMILO; SILVA, 2009).

- **Associação** - Visa identificar regras associativas que relacionem conjuntos de atributos que ocorrem em conjunto de maneira frequente, podendo ser modelado por expressões da forma: se X então Y, ou $X \rightarrow Y$, onde X e Y são conjuntos disjuntos (CORRÊA; SFERRA, 2003).

- **Modelo de Dependência** - Identifica um modelo que descreve dependências significativas entre variáveis, sendo estas definidas em dois níveis: estrutural

e quantitativo. No nível estrutural, o modelo especifica quais variáveis são localmente dependentes entre si. No nível quantitativo, é definido o grau de dependência utilizando escalas numéricas (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Essas tarefas podem ser realizadas por meio da utilização de diversas técnicas e algoritmos, tais como: Redes Neurais, Árvores de Decisão e Regressão Logística, que são categorizados em: aprendizado supervisionado e não supervisionado, além de uma variação entre os dois. No primeiro, existe uma variável alvo, que será utilizada como parâmetro para a categorização de novos registros. Já no não-supervisionado, os modelos são gerados sem que haja uma pré-definição a ser seguida, devendo as categorizações serem realizadas de acordo com as características intrínsecas dos dados, normalmente utilizando medidas de similaridade entre os atributos (CAMILO; SILVA, 2009).

2.2 Web Scraping

A ampla adoção dos dispositivos móveis nas últimas décadas, somada aos significativos avanços nos *hardwares* e *softwares* disponíveis, tem gerado grande impacto no crescimento dos dados (MARQUESONE, 2016). Em 2012, já se estimava que o volume produzido dobrava a cada 18 meses (TAURION, 2012), e para 2025 as previsões já alcançam 175 zettabytes[1] (COUGHLIN, 2018). Nesse cenário de rápida disponibilização de dados e de crescimento acelerado, as redes sociais têm desempenhado um papel crucial, modificando drasticamente a forma como as pessoas se relacionam em uma era cada vez mais digital.

Na nona versão de seu infográfico anual, "*Data Never Sleeps*"[2], a empresa norte-americana Domo[3] traz algumas estatísticas a respeito da quantidade de dados gerados diariamente a cada 1 minuto. No infográfico, mostrado na figura 2.3, é possível

[1] 1 zettabyte = $1 * 10^{21}$ bytes
[2] <www.domo.com/news/press/domo-releases-ninth-annual-data-never-sleeps-infographic>
[3] <www.domo.com>

visualizar que somente no *Facebook*[4] a quantidade de fotos compartilhadas já atinge a marca de 240 mil. James (2021), CEO da Domo, ressalta:

> Não é surpresa como o tipo de modernização e mudança que vimos acontecer no local de trabalho está aparecendo em todos os outros lugares, como na forma como jantamos, como socializamos e até como consumimos entretenimento. (...)
>
> Com a ascensão do trabalho remoto, por exemplo, fizemos de nossas casas nossos escritórios, conhecemos espaços de reuniões digitais como o Zoom como nunca antes e enchemos nossos telefones com aplicativos que nos permitiriam realizar tarefas comuns no trabalho e em casa, borrando ainda mais as linhas entre tecnologia corporativa e tecnologia de consumo (JAMES, 2021, tradução nossa).

Observando as mudanças trazidas pelos novos paradigmas, muitas empresas já identificaram a importância que os dados possuem para a geração de *insights* e tomada de decisões, levando à adoção de estratégias para a captura e o armazenamento em larga escala.

As estratégias de captura apresentam, em sua maioria, abordagens que se beneficiam do uso de APIs (*Application Programming Interfaces*). Estas podem ser definidas como interfaces que podem ser utilizadas, por múltiplos clientes — externos ao ambiente de uma organização —, como ponto de acesso para entidades reutilizáveis de um *software* (ROBILLARD et al., 2012), permitindo a utilização de recursos sem a necessidade de expor a forma como estes são tratados internamente, podendo, inclusive, ser implementadas em linguagens diferentes.

Embora as APIs se mostrem excelentes ferramentas para acessar e integrar os mais diversos ambientes, elas nem sempre se encontram disponíveis ou apresentam as características necessárias para atender uma demanda específica. Fatores como a falta de manutenção por parte da entidade proprietária (ZIBRAN; EISHITA; ROY, 2011), falta de documentação do conhecimento necessário para utilizá-las

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

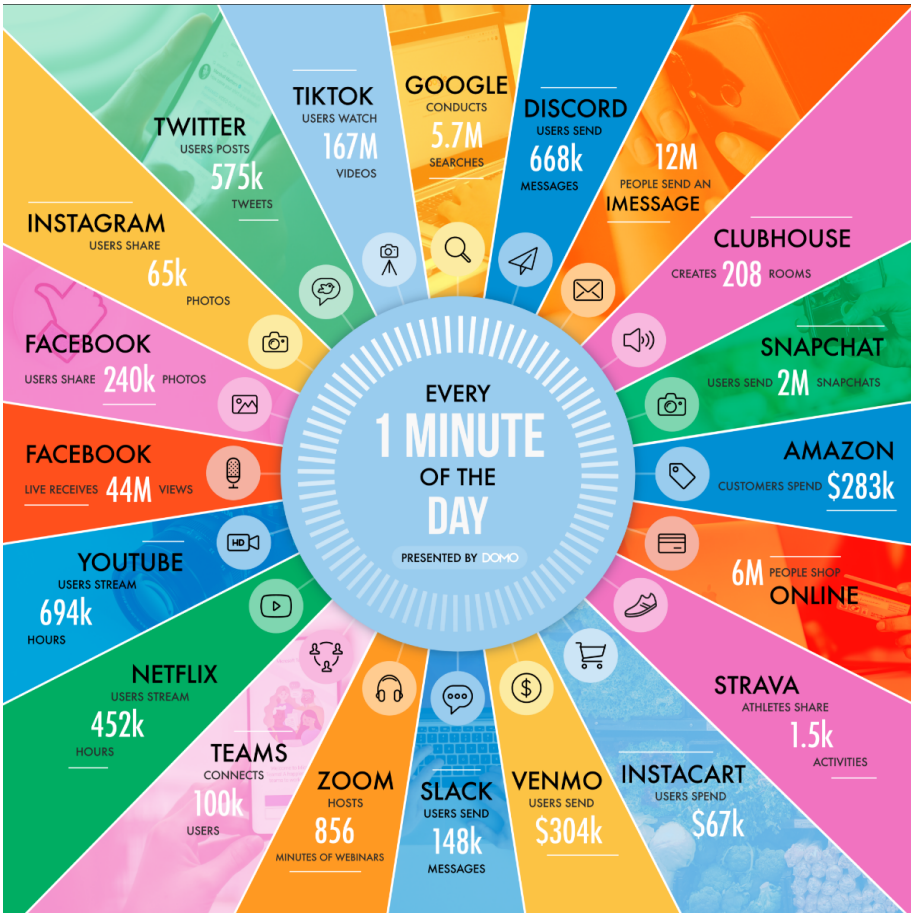

Figura 2.3: Infográfico “Data Never Sleeps” adaptado

propriamente (ROBILLARD et al., 2012) e limitações de acesso são exemplos dos possíveis problemas de uma abordagem que utilize APIs.

Em meio aos fatores apresentados, a técnica de *Web Scraping* se apresenta como uma alternativa razoável ao uso de APIs, sendo capaz de abranger uma vasta quantidade de informações disponíveis na internet, uma vez que estas se encontram, em sua maioria, dispostas em páginas HTML (*HyperText Markup Language*) e, portanto, podem ser tratadas via varreduras em suas estruturas.

De maneira geral, *Web Scraping* — também conhecido como *screen scraping* ou *web harvesting* — pode ser definido como uma técnica para extração de dados, normalmente não estruturados, de páginas da web, com o objetivo de gerar um conjunto de dados estruturados que atendam a algum interesse específico, podendo ser exportados em um formato de fácil manipulação como CSV (*Comma Separated Values*) ou armazenados em um banco de dados para análise posterior (ZHAO, 2017).

Esse procedimento tem sido utilizado para diferentes propósitos tais como: monitoramento de preços pelo mercado, geração de *leads* para campanhas publicitárias, captação de contatos, pesquisas e monitoramento de marca (KHDER, 2021).

Segundo Zhao (2017, tradução nossa), o processo de *Web Scraping* pode ser dividido em duas etapas sequenciais "adquirir recursos da web e, em seguida, extrair as informações desejadas dos dados adquiridos". Nessa visão, o *scraper* deve realizar uma requisição HTTP (*HyperText Transfer Protocol*) para a página desejada e realizar o *download* de seu conteúdo, devendo em seguida realizar os tratamentos necessários para organizá-lo em um formato estruturado (ZHAO, 2017).

Saurkar, Pathare e Gode (2018) comentam que, de um ponto de vista operacional, o processo de coleta de dados via *web scraper* muito se assemelha com um típico movimento de copiar e colar realizado por humanos no dia a dia, diferenciando-se pela capacidade de realizar múltiplas solicitações simultâneamente e de seguir um fluxo automatizado de forma uniforme e ininterrupta. Essa capacidade, no entanto, pode gerar algumas complicações se utilizada indiscriminadamente, uma vez que as sucessivas requisições realizadas nas páginas nas quais os dados se encontram podem ser interpretadas como uma tentativa de negação de serviço (ou em inglês, *Distributed Denial of Service - DDoS*).

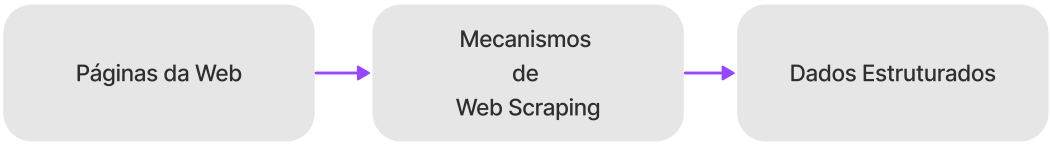

Figura 2.4: Estrutura Base de Web Scraping. Adaptado de Saurkar, Pathare e Gode (2018)

Em busca de evitar os problemas causados pelo uso indiscriminado das técnicas de *scraping*, grande parte das plataformas impõem restrições nos termos de uso de suas páginas, que devem ser observadas por ferramentas de captura, mantendo a frequência de requisições sob controle. Para tratar o uso indesejado de suas páginas, as plataformas fazem uso de diversas técnicas, podendo realizar a identificação dos

visitantes e realizar o mapeamento do histórico do comportamento, observando padrões anormais tais como altas taxas de requisições e navegações suspeitas (ZHAO, 2017). A adoção de práticas de visitação incompatíveis com os termos de uso das plataformas pode resultar na restrição do acesso a determinadas seções, até o banimento do usuário autenticado ou do próprio endereço IP, podendo o banimento ter caráter temporário ou até permanente.

Capítulo 3

Proposta

Este capítulo descreve uma proposta de uma ferramenta para a coleta de informações no Instagram, apresentando sua motivação e os trabalhos relacionados.

3.1 Motivação

As redes sociais têm ganhado espaço cativo na vida das pessoas ao longo dos anos, eliminando distâncias geográficas, criando comunidades e levando acesso a assuntos diversos para os mais variados grupos sociais. Com formato voltado majoritariamente para mídias, o Instagram tem conquistado público em ritmo bastante acelerado, ultrapassando a marca de um bilhão de usuários (SMITH, 2019), agradando não só os usuários típicos das redes concorrentes, mas também pessoas que não conseguiam ter a sensação de pertencimento em meio às outras redes. Somente no Brasil, a rede já ultrapassava a marca de 119 milhões de usuários em Janeiro de 2022, colocando o pais na terceira colocação no *ranking* mundial (DIXON, 2022), conforme mostrado na figura 3.1.

Ao dar destaque às mídias, o Instagram criou uma nova forma de comunicação entre os usuários, o que rapidamente gerou impacto nas abordagens de captação de consumidores por parte das marcas anunciantes, tornando a comunicação mais direta e informal, passando a ideia de proximidade com o público alvo. Esse tom

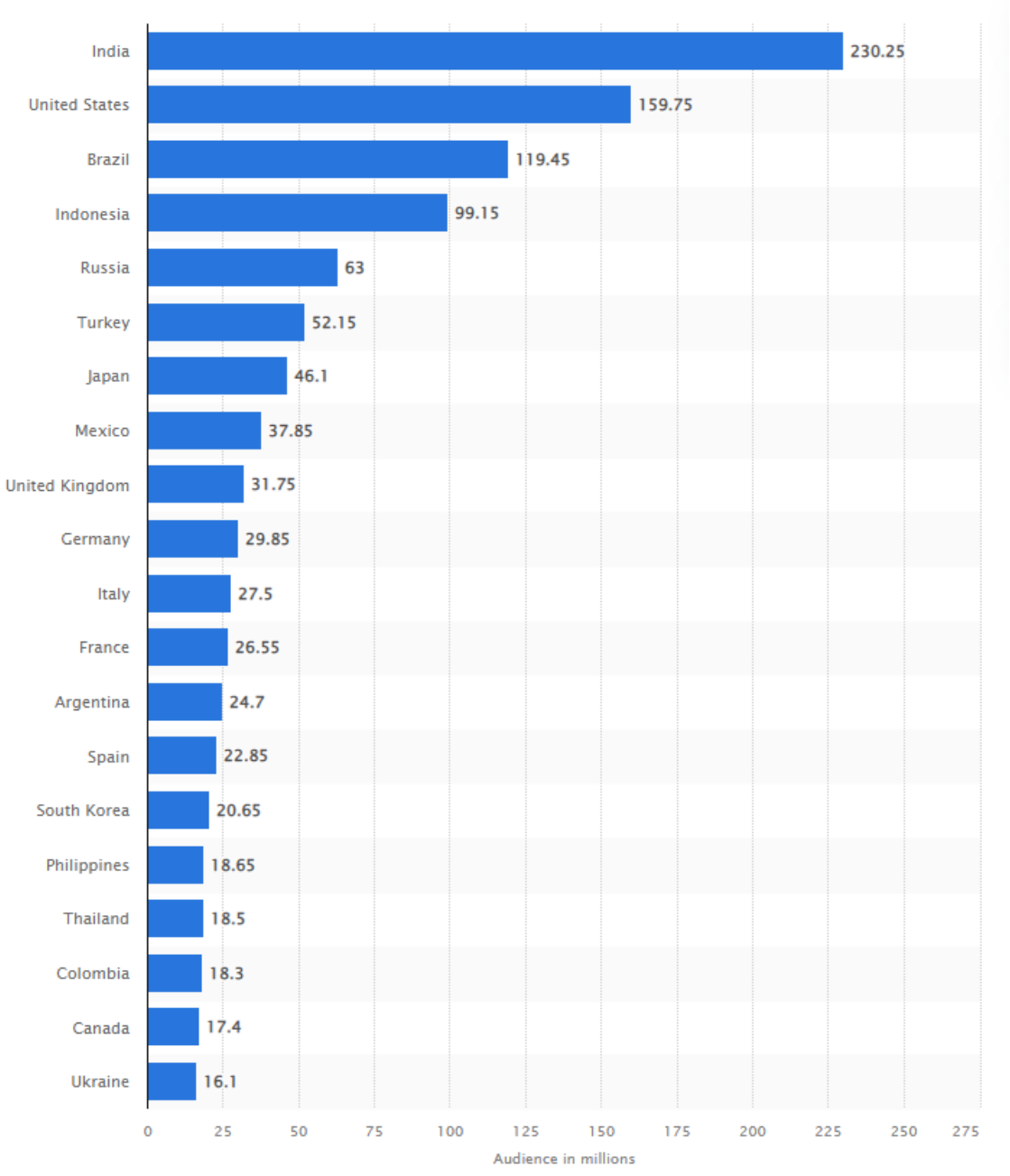

Figura 3.1: Países com maior utilização do Instagram em Janeiro de 2022, retirado de (DIXON, 2022)

mais informal também gerou o impulso nos comunicadores locais/não institucionais, gerando um ciclo de tendências, angariando mais usuários, que por sua vez demandam cada vez mais presença no meio digital, reduzindo consideravelmente os intervalos entre postagens.

Se por um lado o ambiente virtual possibilita um “encurtamento das distâncias”

e uma maior facilidade de comunicação por conta de sua natureza assíncrona, por outro a constante exposição à realidades distintas e a categorização e seleção do que deve ser mostrado, e a quem deve ser mostrado, por parte de algoritmos resulta em uma alteração na forma com que as pessoas avaliam o próprio "sucesso social", levando a uma valorização de parâmetros como: número de *likes*, compartilhamentos, comentários e visualizações.

Outro aspecto trazido pela nova forma de socialização é a construção, por vezes involuntária e inconsciente, de um histórico social do indivíduo, contemplando inúmeras informações a seu respeito tais como: o que curte, com quem conversa, sobre o que conversa, locais que frequenta, entre outras. Toda essa profundidade de informações disposta nas redes sociais, e a possibilidade do cruzamento de dados de outros indivíduos faz com que plataformas como Instagram, Twitter e Facebook se tornem excelentes ferramentas para o mapeamento da forma de pensar e agir de um coletivo.

Nesse contexto, a riqueza de informações contidas na rede possibilita o desenvolvimento de diversas frentes de estudo, abordando os mais diversos tópicos, fornecendo informações suficientes para identificação de padrões, que podem ir de simples formas de edição de imagens à análise de sentimentos utilizando textos de comentários ou tratamento de imagens e verificação de tópicos com maior interação.

Embora a imensidão de dados gerados pelas redes sociais possa ser visto como um ambiente altamente rico em informações, muito propício para pesquisas, extração de conhecimento e geração de *insights*, seu acesso ainda ocorre de forma bastante limitada e altamente dependente da disponibilização desses dados por meio de APIs fornecidas pelas próprias plataformas.

A disponibilização de API, no entanto, não é uma garantia, uma vez que em muitos casos, não há interesse por parte da entidade proprietária em disponibilizar um fácil acesso às informações ou em manter a estrutura necessária para o correto funcionamento da API, tornando a captura de dados mais dificultosa.

Alternativamente, a captura pode ser realizada por meio da utilização de *web*

scrapers, simulando o comportamento de um usuário típico da plataforma, sendo capaz de navegar por todas as funcionalidades disponíveis e possibilitando a realização de atualizações de forma independente. Dessa forma, futuras alterações de funcionalidades e novas informações contempladas na rede social podem ser tratadas sem a dependência de atualizações por parte da origem das informações.

Muitos dos desafios para a captura de dados podem ser explicados pela própria natureza das plataformas, que pode ser entendida como uma fusão entre o público e o privado. Se por um lado, as publicações dos usuários e suas interações são realizadas e expostas de forma pública, podendo ser visualizada e interpretada por qualquer pessoa, a forma como essas informações são utilizadas e o grau com que são alcançadas pelo mundo externo são regulados de acordo com as políticas das plataformas onde estão inseridas, que, em muitos casos, implementam formas de dificultar a coleta de dados em escala.

Um exemplo das limitações impostas pode ser verificado pela adoção de cláusulas nos termos de uso, ressaltando que a permanência do usuário na plataforma é condicionada ao atendimento de suas politicas de utilização, mencionando que a coleta de dados em suas páginas de domínio pode resultar na expulsão do usuário.

3.2 Trabalhos Relacionados

O crescimento do número de usuários do Instagram e o comportamento integrado entre as redes sociais por parte dos usuários tem chamado a atenção de pesquisadores, fazendo com que a rede fosse incluída no escopo de trabalho. Muitos destes se restringem às análises das informações coletadas, não entrando em detalhes sobre a forma de captura. Já nos trabalhos que abordam a metodologia de coleta as formas vão desde o uso de *scrapers* à utilização de APIs do Instagram, que eram disponibilizadas anteriormente, incluindo abordagens híbridas.

Zarei, Farahbakhsh e Crespi (2019) desenvolveram um *crawler multi-threaded* para coletar dados do Instagram e armazenar em banco utilizando MongoDB[1]. Fazendo

[1]<https://www.mongodb.com/pt-br>

uso da API do Instagram, eles dividiram o processo de coleta em 4 módulos: *post*, *comment*, *like* e *profile*, que ficavam responsáveis por requisitar à API as informações de acordo com o escopo definido. O trabalho, no entanto, não especifica em mais detalhes como o processo é realizado e como é feito o gerenciamento entre os módulos.

Carta et al. (2020) realizaram a captura separando as atividades em etapas. Em um primeiro momento, os autores selecionaram uma lista preliminar de usuários do Instagram e, em seguida, utilizaram uma extensão do Google Chrome[2] para coletar a lista completa de seguidores desses perfis. De posse dessa lista, utilizaram uma aplicação desenvolvida em python para remover duplicatas e realizar uma filtragem na lista coletada, limitando-a a usuários com menos de 25 mil seguidores, com pelo menos 100 postagens e descartando perfis privados. Com a lista final, utilizaram um projeto *open-source* para realizar a captura dos dados efetivamente, salvando os resultados em um arquivo JSON.

Já Himawan, Priadana e Murdiyanto (2020) estruturaram o processo de extração iniciando por uma fase de análise, em que o pesquisador deve analisar a estrutura HTML e JSON da página do Instagram e determinar quais elementos devem ser coletados. Em seguida, utilizando a biblioteca python Beautiful Soup, deve realizar uma requisição para a página do Instagram, para coletar as informações em JSON, separando os atributos desejados. Os autores desenvolveram uma aplicação separada em 3 partes: *Media Crawler*, que fica responsável por realizar a coleta de dados; *Data Repository*, que consiste na utilização de MongoDB para armazenar os dados coletados; e *Web User Interface*, que consiste em uma interface desenvolvida em Flask que permite ao usuário gerenciar os dados coletados, podendo exportá-los para CSV, Excel ou JSON. Os autores ainda mencionam que nos testes realizados foi encontrado o valor de 2412 como limite para *download* de publicações por conta do Instagram.

De forma alterativa, a APIFY[3] se apresenta como uma plataforma para automação e *scraping* na web, disponibilizando ferramentas ("*Actors*") para cada propósito

[2] <https://www.google.com/intl/pt-BR/chrome/>
[3] <https://apify.com/>

específico. No caso do "*Actor* para Instagram"[4] em especial, é disponibilizado a extração a partir de *profiles*, *hashtags* e *locations*, se denominando como "uma API não oficial para o Instagram desenvolvida para devolver a funcionalidade de acessar dados públicos que fora removida da API do Instagram em 2020". Para contornar as restrições impostas pelo Instagram, a plataforma faz uso de *proxies* residenciais que, por sua vez, adicionam custo à operação.

Outra opção fica por conta do projeto *open-source* InstagramCrawler[5], que se apresenta como uma ferramenta para a captura de dados do Instagram, disponibilizando opções para coleta de informações de perfis, postagens e seus comentários associados, e coletas a partir de *hashtags*, além de disponibilizar uma opção para realizar a "curtida" automática de postagens. O projeto, no entanto, apresenta problemas, uma vez que não acompanhou as modificações realizadas pela rede social, fazendo com que a maioria de suas funcionalidades deixem de funcionar corretamente.

3.3 Proposta

Sua natureza voltada para o compartilhamento de mídias, somada à ausência de suporte oficial para a disponibilização de dados abrangentes para pesquisadores, têm limitado os trabalhos utilizando o Instagram como fonte de estudo, em detrimento de outras redes com maior riqueza textual e facilidade de obtenção de dados, como: Twitter, Telegram e WhatsApp.

O Instagram pode ser visto como a modelagem apresentada na imagem 3.2, onde estão dispostas as relações elementares realizadas entre perfis na rede social. A partir do modelo, é possível verificar a possibilidade de mapear o comportamento de um usuário na rede, identificando os comentários realizados e quais tipos de conteúdo despertam algum tipo de reação, que pode ser verificado a partir de comentários, reações de "curtir" e atos do tipo "seguir".

Tendo em vista o cenário apresentado, somado ao crescimento de estudos utili-

[4] <https://apify.com/jaroslavhejlek/instagram-scraper#features>
[5] <https://github.com/huaying/instagram-crawler>

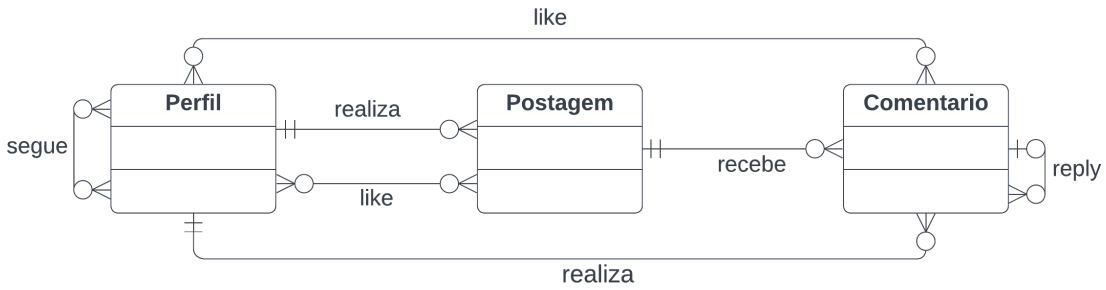

Figura 3.2: Representação das relações elementares realizadas no Instagram

zando imagens e a necessidade de um entendimento do comportamento social na internet de forma abrangente, torna-se extremamente relevante a abordagem de estudos em direção ao Instagram. Para que tais estudos sejam viabilizados, é necessário a disponibilização de alternativas para captura de informações na plataforma, de maneira que esta possa ser utilizada de forma independente de atualizações e permissões por parte do Instagram, e moldada à medida que a plataforma apresenta novas funcionalidades aos usuários, possibilitando a observação dos padrões de utilização e um controle maior por parte do pesquisador.

Entre os modelos de captura, uma abordagem modularizada possibilitaria uma resposta mais rápida às alterações realizadas na plataforma — quando comparada com os modelos verticalizados —, gerando maior robustez na ferramenta como um todo, além de possibilitar o tratamento de pontos específicos por seus maiores interessados, que tendem a apresentar um resultado mais completo quando comparado a um desenvolvimento generalizado.

Com o intuito de atingir o objetivo mencionado, este trabalho se propõe a desenvolver uma ferramenta de captura independente, dividida em diferentes serviços de acordo com o item a ser observado e capturado, além de uma API isolada, responsável por fazer a conexão com o banco de dados e repassar quais operações devem ser realizadas por cada serviço autônomo. A separação em serviços possibilitará que um usuário, à medida que julgue necessário, faça alterações apenas nos módulos que tratem de sua área de interesse, os atualizando e mudando seus comportamentos para contemplar uma abordagem específica ou uma nova funcionalidade disponibilizada

pelo Instagram.

A partir da estrutura definida, o usuário deve ser capaz de, dado um perfil alvo, capturar as postagens realizadas pelo perfil, sua URL de identificação na rede, sua legenda atribuída, a data em que a postagem foi realizada e a lista de comentários associados. Para cada perfil, também deve ser possível a obtenção da lista de perfis que seguem e que são seguidos pela conta. Além disso, o usuário deverá ser capaz de obter a lista de pessoas que clicaram no botão de curtir para cada postagem observada. Todas essas informações deverão ser armazenadas de maneira a facilitar uma futura compilação de dados e a realização de análises.

Para que o usuário possa ter um controle maior na definição de qual perfil deverá ser trabalhado pelo sistema, será fornecida a possibilidade de dar prioridade ao tratamento de um perfil específico. Essa prioridade será definida de acordo com o escopo de cada serviço, permitindo dar maior ênfase ao conjunto de informações que realmente interessa e que possua uma necessidade de atualização mais específica. Nesse contexto, também será permitida a adição de perfis numa *watchlist* que definirá uma frequência de atualização para os perfis nela inscritos.

No sistema proposto, os perfis são classificados em dois grupos: **perfis visitados** e **perfis não visitados**. Os primeiros são caracterizados por já terem percorrido ao menos um ciclo de captura pelos serviços que compõem o sistema, sendo atribuído uma descrição contendo as informações biográficas do perfil, suas postagens, lista de seguidores ou comentários associados. Já os perfis não visitados remetem a usuários meramente identificados como pertencentes ao ambiente do Instagram, não tendo ocorrido nenhum preenchimento de sua estrutura de informações. Estes, ao serem identificados durante uma captura de **lista de seguidores, *likers* ou comentários**, são adicionados ao banco de dados do sistema e marcados para futura visitação, possibilitando a expansão da rede de conexões mapeadas pela ferramenta.

Capítulo 4

Implementação

Visando atingir os objetivos descritos na seção 3.3, o presente trabalho propõe a criação de um sistema em que o usuário poderá adicionar uma conta do *Instagram* e realizar a coleta de dados. Tal sistema poderá ser instalado em um computador doméstico simples ou em plataformas de IaaS (*Infrastructure as a Service*) com gerenciamento de carga e fácil escalonamento.

Nas seções deste capítulo será possível encontrar as decisões técnicas que foram tomadas ao longo do projeto, além da divisão do sistema em módulos, alguns exemplos de utilização e os resultados obtidos.

4.1 Tecnologias Utilizadas

Uma das grandes dificuldades enfrentadas no desenvolvimento de qualquer projeto é a definição das tecnologias a serem utilizadas. Para esta tarefa, questões como facilidade de manipulação e curva de aprendizado devem ser observadas, influindo diretamente na capacidade de evolução e no sucesso de um projeto.

Visando facilitar o desenvolvimento do sistema proposto e suas futuras atualizações, as escolhas tomadas neste trabalho consideraram a adoção das tecnologias utilizadas pelo mercado, partindo do principio que a probabilidade de um usuário dominar as ferramentas utilizadas é mais alta para tecnologias amplamente

difundidas.

Nesse contexto, para o desenvolvimento do projeto apresentado, foram utilizados: a linguagem Python, fazendo uso do Selenium para a interação com o *browser*; o *framework* Flask[1], para o desenvolvimento da API responsável pelo controle do sistema; o SGBD (Sistema Gerenciador de Banco de Dados) PostgreSQL para armazenamento dos dados; o ORM (*Object-Relational Mapping*) Peewee[2] para a integração entre o sistema e o banco de dados; e Docker[3] para a separação do sistema em contêineres.

4.2 Implementação do Projeto

A implementação deste trabalho tomou como ponto de partida o projeto *InstagramCrawler*[4], que já apresenta um conjunto de funcionalidades pré-implementadas e uma estrutura monolítica com o comportamento de um *scraper*. Apesar de levar o nome de *crawler*, o projeto não realiza o tratamento para captura contínua e não apresenta tratamento de erros, o que pode ser um empecilho para um processo de longa duração e que depende de uma capacidade de execução de forma automatizada e com acompanhamentos esporádicos.

O uso do projeto como ponto de partida tem como justificativa o fato de sua implementação ser feita em *python* com uso do *Selenium*, o que facilita sua utilização uma vez que possuem sintaxe de simples entendimento e que é amplamente difundida, o que possibilitaria uma futura atualização por parte de um usuário que deseje implementar uma nova funcionalidade ou ampliar uma existente.

Apesar dos benefícios trazidos pelo *InstagramCrawler*, ainda foi necessário a realização de uma série de adaptações, uma vez que o projeto original apresentava funções deficitárias e desatualizadas com relação a estrutura atual do *Instagram*, tornando necessário o remapeamento de *tags* utilizadas no *HTML* da rede social, e

[1] <https://flask.palletsprojects.com/en/2.1.x/>
[2] <http://docs.peewee-orm.com/en/latest/>
[3] <https://www.docker.com/>
[4] <https://github.com/huaying/instagram-crawler>

a reestruturação da forma de operação do projeto para que este se comportasse de forma mais condizente com as limitações do modelo utilizado, e mais otimizada de acordo com as políticas da rede social.

Além dos pontos já mencionados, o projeto também apresentava uma forte dependência de ações por parte do usuário, sendo necessário que este forneça um perfil a ser tratado para cada nova execução do projeto, não possibilitando um crescimento natural por meio da observação das interações realizadas entre os diferentes perfis identificados na captura original.

Visando atingir os objetivos propostos no capitulo 3, o sistema foi estruturado em serviços de forma a possibilitar um escalonamento de acordo com a necessidade de cada faixa de interesse, sendo os serviços divididos em cinco tipos: *instagram-crawler-description*, *instagram-crawler–follow*, *instagram-crawler–likers*, *instagram-crawler–post* e *instagram-crawler–post-downloader*.

Uma vez iniciado, cada serviço realizará uma chamada para API para obtenção de uma lista de pontos de partida, podendo ser: uma URL para uma postagem, um perfil a visitar, ou uma URL de mídia a ser baixada, variando de acordo com o tipo de serviço requisitante, conforme mostrado na figura 4.1.

instagram-crawler–follow:

Responsável pela identificação das relações entre usuários do Instagram, o serviço *follow* deverá, de posse de um perfil alvo, realizar a navegação na plataforma do Instagram até alcançar o seu ponto de partida, devendo a partir deste, selecionar a seção de seguidores, realizando o rolamento do menu interativo, capturando os perfis pertencentes à lista, até que seu critério de parada seja atingido. De posse da lista de perfis, o serviço deve adicioná-los ao banco de dados como perfis não visitados, para que seja possível identificar um usuário do Instagram no ecossistema do software aqui apresentado, e marcando-o para futuras visitas, a fim de completar as informações faltantes, alterando o seu estado de "não visitado" para "visitado".

instagram-crawler–likers:

Responsável pela obtenção da lista de pessoas que reagiram a uma postagem

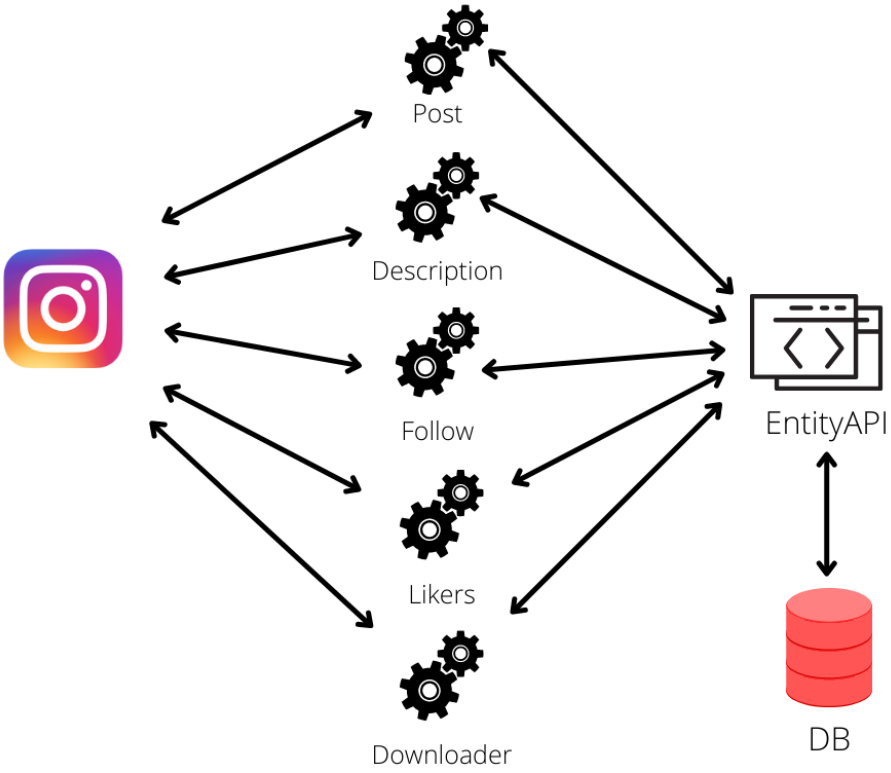

Figura 4.1: Representação da divisão do sistema em serviços

por meio do botão de gostei no Instagram, o serviço *likers* deverá, de posse de uma postagem alvo, realizar a navegação até a URL da postagem e, a partir dessa, selecionar a seção de "gostei" e realizar o rolamento do menu iterativo até que seu critério de parada seja atingido, capturando os perfis pertencentes à lista. De posse da lista de perfis, tal como o serviço *follow*, deverá efetuar a persistência dos dados coletados, possibilitando o mapeamento das interações com a postagem e a descoberta de novos perfis a serem visitados.

instagram-crawler–post:

Responsável pela captura de *posts* pertencentes a perfis do Instagram, o serviço *post* deverá, de posse de um perfil alvo, realizar a navegação na plataforma do Instagram até alcançar o seu ponto de partida, devendo, a partir deste, realizar o rolamento da lista de *posts* e capturar as informações destes como legenda, número de curtidas e URLs das mídias associadas. De posse das informações dos *posts*, o serviço deve adicioná-las ao banco de dados, possibilitando o mapeamento do *feed* do perfil e funcionando também como fonte de URLs das mídias associadas a serem

usadas pelo serviço *post-downloader*.

instagram-crawler–post-downloader:

Responsável pelo descarregamento das mídias associadas a *posts*, o serviço *post-downloader* deverá, de posse de uma URL de mídia, realizar o *download* da mídia, que pode ser uma imagem ou vídeo. Com a posse desses dados, o serviço deve adicioná-los ao banco de dados a fim de ser utilizado pelo usuário do sistema para um propósito geral.

instagram-crawler–description:

Responsável pela geração de um "retrato" de um perfil num determinado momento, o serviço *description* deverá, de posse de um perfil alvo, realizar a navegação na plataforma do *Instagram* até alcançar o seu ponto de partida, devendo, a partir deste, capturar as informações que caracterizam o perfil, como: nome do usuário na rede, descrição de dados biográficos, informações de contato, foto identificadora do perfil, numero de postagens realizadas até o momento e quantidade de pessoas nos atributos "seguindo" e "seguidores". De posse dessas informações, o serviço deverá realizar a atualização do perfil no banco de dados, possibilitando uma visão completa a respeito de um perfil.

4.2.1 Lista de Prioridade e *Watchlist*

Para a implementação da priorização de perfis e geração de *watchlist*, foi gerado um mecanismo de controle, que ficará responsável por monitorar, a cada geração de uma nova lista de trabalho a ser retornada para os serviços, os parâmetros definidos pelo usuário e que devem ser atendidos pelo escalonador de trabalho. Para tal, foram geradas duas tabelas no banco de dados: *PriorityList* e *Watchlist*, conforme a figura 4.2. Nelas, o campo *serviceIdentifier* será utilizado para identificar o serviço para o qual o usuário apresenta prioridade, e o campo *expirationDate* será utilizado para a identificação de quais itens de atualização periódica já estão aptos a serem trabalhados.

Com o intuito de manter os serviços escaláveis, de maneira isolada do controle de

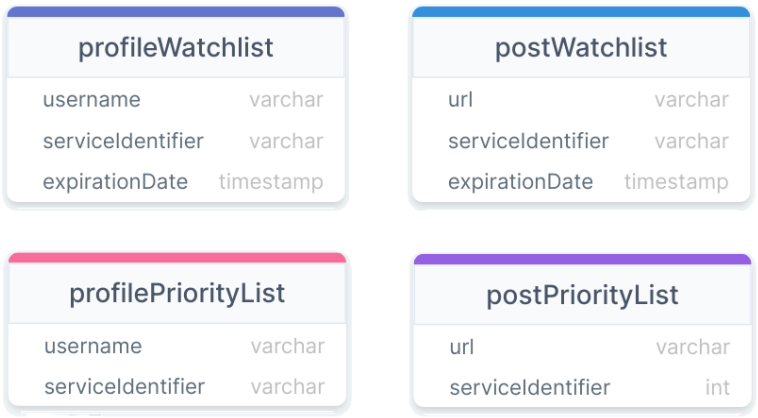

Figura 4.2: Tabelas de Controle *PriorityList* e *Watchlist*

suas listas de trabalho, foi decidido que esta responsabilidade seria atribuída à API, ficando esta encarregada de mesclar a lista de trabalho padrão de cada serviço com as listas priorizadas e as de atualização periódica. À API também foi atribuída a função de gerar os itens na tabela de controle das *watchlists* e das priorizadas, de acordo com as demandas por cada serviço, realizando a atualização dos campos de *expirationDate* e removendo os itens da tabela de prioridades ao final de cada ciclo de trabalho realizado pelos serviços.

4.2.2 *EntityAPI*

De forma a possibilitar o escalonamento dos diferentes serviços, de acordo com as necessidades do usuário, foi decidido que a comunicação do sistema se daria por meio de um centralizador que ficaria responsável pelo controle dos diferentes serviços, atribuindo a estes as atividades a serem desempenhadas e abstraindo a gestão dos dados a serem mantidos. Dessa forma, foi criada a *EntityAPI*, que atua como ponte entre o banco de dados e os diferentes serviços, além de intermediar a comunicação do usuário com o sistema.

Nessa abordagem, para que um serviço possa realizar suas atribuições deve ser efetuada uma sequência de etapas. Iniciando por uma chamada para a *EntityAPI*, o serviço deve obter uma lista de trabalho, que conterá uma sequência de itens a partir dos quais a coleta deve ser realizada, podendo ser um *username* do *Instagram* para os serviços que têm como ponto de partida um perfil, ou a *url* de uma postagem

pros serviços que efetuem suas coletas a partir dessas. Após a realização de uma captura, o serviço deve realizar uma nova chamada para a API, passando os dados coletados e o tipo de serviço a partir do qual a coleta foi realizada, para que a API possa realizar o controle dos dados e dos contextos de atualização.

Para a definição da lista de trabalho a ser passada para o serviço, a *EntityAPI* realiza uma junção de três listas distintas: lista padrão, lista prioritária e *watchlist*, conforme o diagrama de sequência demonstrado na figura 4.3. Para a geração da lista padrão, a API faz uso de um fator de proporcionalidade, que é um parâmetro definido para realizar o controle da quantidade de perfis visitados e perfis não visitados que deve ser passada para o atual ciclo de trabalho do serviço. É por meio desse fator que o sistema realiza o controle da taxa de crescimento de seu universo de domínio e sua atualização.

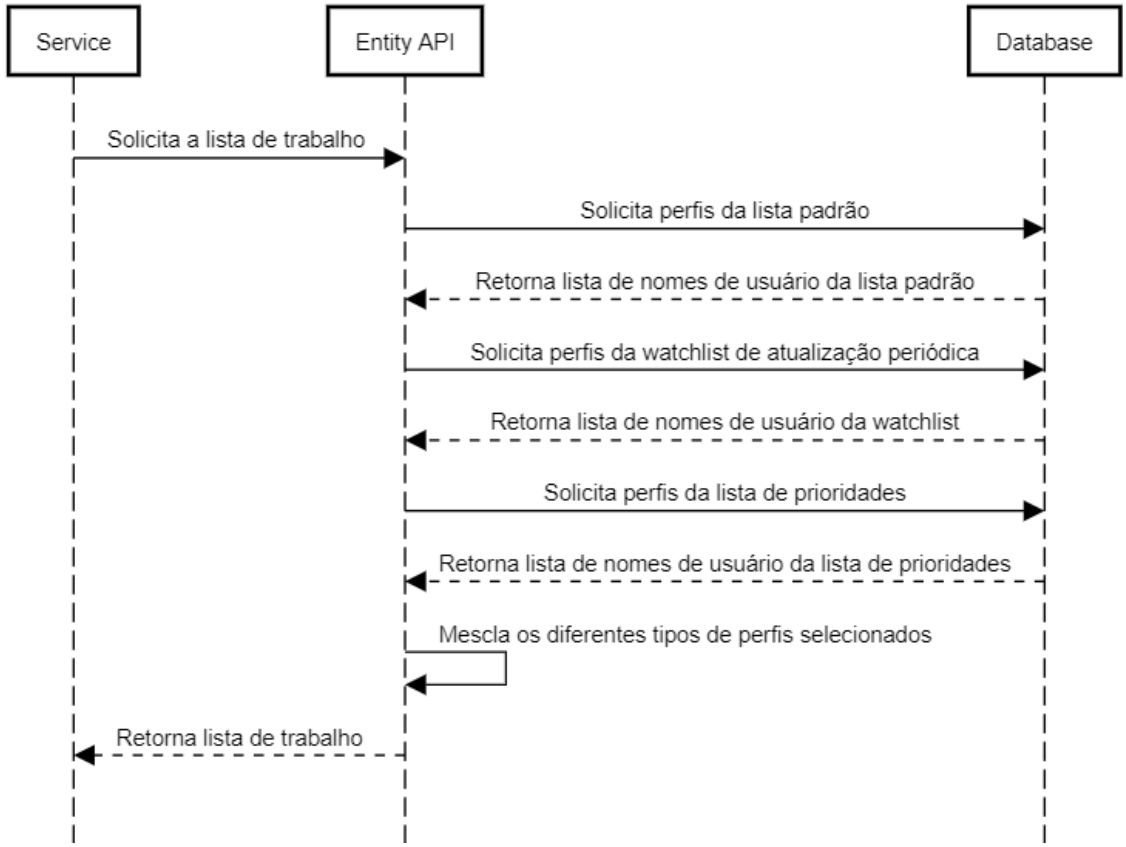

Figura 4.3: Diagrama de Sequência da geração da lista de trabalho de serviços que utilizam *username*

Já para a lista priorizada, a *EntityAPI* verifica a existência de itens priorizados

para um serviço por meio da tabela de controle, utilizando como parâmetro o nome do serviço para o qual a lista se destina, adicionando os itens encontrados na lista de trabalho e os removendo da tabela de controle de prioridade. No caso da *watchlist*, a verificação da existência de itens de trabalho é realizada por meio do campo *expirationDate* — que funciona como um sinalizador de que o item já pode ser atualizado, devendo ter seu valor editado sempre que o item sofre uma atualização —, além do nome do serviço para os quais os itens se destinam.

O desenvolvimento da API foi realizado utilizando o Flask[5], um micro framework para a criação de aplicativos web em python, tornando a implementação da API REST mais fácil, uma vez que este já traz o arcabouço necessário para realizar o mapeamento das rotas e apresenta uma estrutura bastante enxuta e de simples utilização, eliminando a necessidade de realizar configurações desnecessárias.

4.2.3 Organização dos Dados no Sistema

Para a organização dos dados no sistema, foi definida a estrutura apresentada na figura 4.4, nela são definidas quatro entidades: *Profile*, *Post*, *Comment* e *Post_ Media*, que são utilizadas para o armazenamento e controle dos diferentes relacionamentos identificados na rede social.

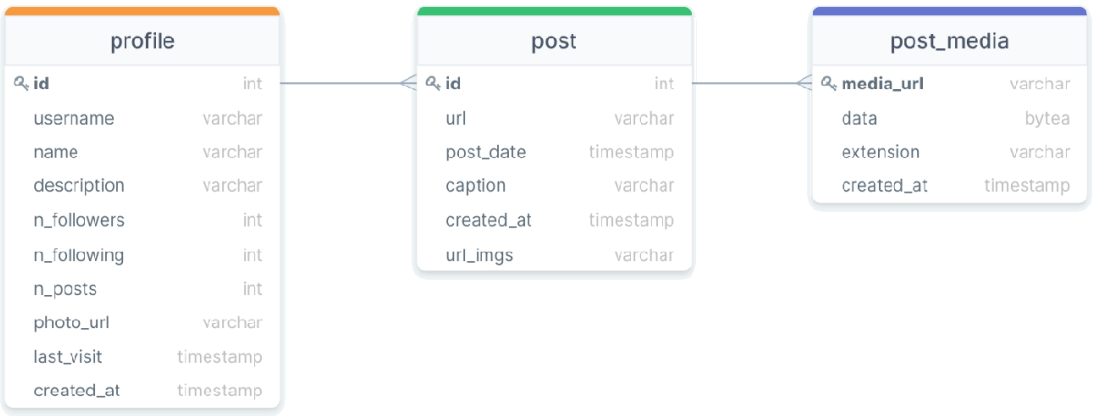

Figura 4.4: Diagrama Entidade Relacionamento - *Instagram*

A entidade *Profile* tem seus atributos preenchidos a partir dos serviços *instagram-*

[5]<https://flask.palletsprojects.com/en/2.1.x/>

crawler-description, *instagram-crawler–follow* e *instagram-crawler–likers*, sendo o primeiro responsável pelo preenchimento dos campos *name*, *description*, *n_followers*, *n_following*, *n_posts* e *photo_url*, e os dois últimos responsáveis pelo preenchimento do campo *username*, apenas, atuando como expansores do universo de domínio do sistema.

Na entidade *Post*, os campos *url*, *url_imgs*, *post_date* e *caption* são preenchidos pelo serviço *instagram-crawler–post*, sendo o campo *url_imgs* utilizado pelo serviço *instagram-crawler–post-downloader* para a identificação da *url* a partir da qual o download deve ser realizado.

Já na entidade *Post_Media*, os campos *media_url* e *extension* são utilizados pelo serviço *instagram-crawler–post-downloader* para identificação da *url* a partir da qual o download foi realizado e do formato que deve ser utilizado para o correto armazenamento do arquivo, respectivamente.

Com o intuito de facilitar o gerenciamento dos dados no sistema e sua manipulação, foi utilizado o *Peewee*, um ORM (*Object-Relational Mapping*) minimalista para a linguagem python que tem como principais características sua simples instalação, podendo ser realizada pelo gerenciador de pacotes pip, e sua fácil utilização, abstraindo as consultas SQL e permitindo a manipulação dos dados a partir de objetos python.

Para que o Peewee consiga realizar o mapeamento corretamente, é necessário fazer a implementação de classes estendidas do ORM —como mostrado na figura 4.5—, que utilizam tipos próprios[6] e que abstraem as diferenças entre os tipos usados pelos SGBDs (Sistemas Gerenciadores de Bancos de Dados), realizando a tradução a partir do modelo definido para uma tabela no banco de dados, tornando, assim, o processo transparente para o sistema e facilitando a utilização e migração entre SGBDs distintos ao longo do ciclo de vida do software.

[6] <http://docs.peewee-orm.com/en/latest/peewee/models.html#field-types-table>

```python
from peewee import AutoField, TextField, IntegerField, BigIntegerField, BooleanField

from persistence.entity.base_model import BaseModel


class ProfileEntity(BaseModel):
    id = AutoField(null=False)
    username = TextField(null=False, unique=True)
    name = TextField(null=True)
    description = TextField(null=True)
    n_followers = IntegerField(null=True)
    n_following = IntegerField(null=True)
    n_posts = IntegerField(null=True)
    photo_url = TextField(null=True)
    last_visit = BigIntegerField(null=True)
    created_at = BigIntegerField(null=True)
    deleted = BooleanField(null=True)
    visited = BooleanField(null=True)
    delay = IntegerField(null=True)

    class Meta:
        table_name = 'profile'

```

Figura 4.5: Modelo utilizado pelo Peewee para a representação da entidade *Profile*

4.3 O processo de captura

Para que o sistema fosse capaz de realizar a coleta das informações a partir da plataforma do *Instagram*, foi necessário uma etapa de análise da estrutura HTML, para a identificação dos pontos de coleta que são trabalhados a partir do *Selenium*. Nessa etapa, ilustrada pela imagem 4.6, utilizando a opção "*DevTools*" disponibilizada no *Google Chrome*, foi observado o conjunto de marcadores utilizados para cada seção do *Instagram*, a fim de gerar as instruções que devem ser passadas para a biblioteca.

Uma vez que todos os marcadores são identificados, é possível iniciar a captura efetivamente. Para tal, é utilizada a biblioteca do *Selenium* para realizar a navegação até a página onde encontra-se a informação desejada, e então, de posse do marcador previamente identificado, realizar a filtragem para que o resultado reflita apenas os elementos alvo. Um exemplo dessa etapa é demonstrado na imagem 4.7.

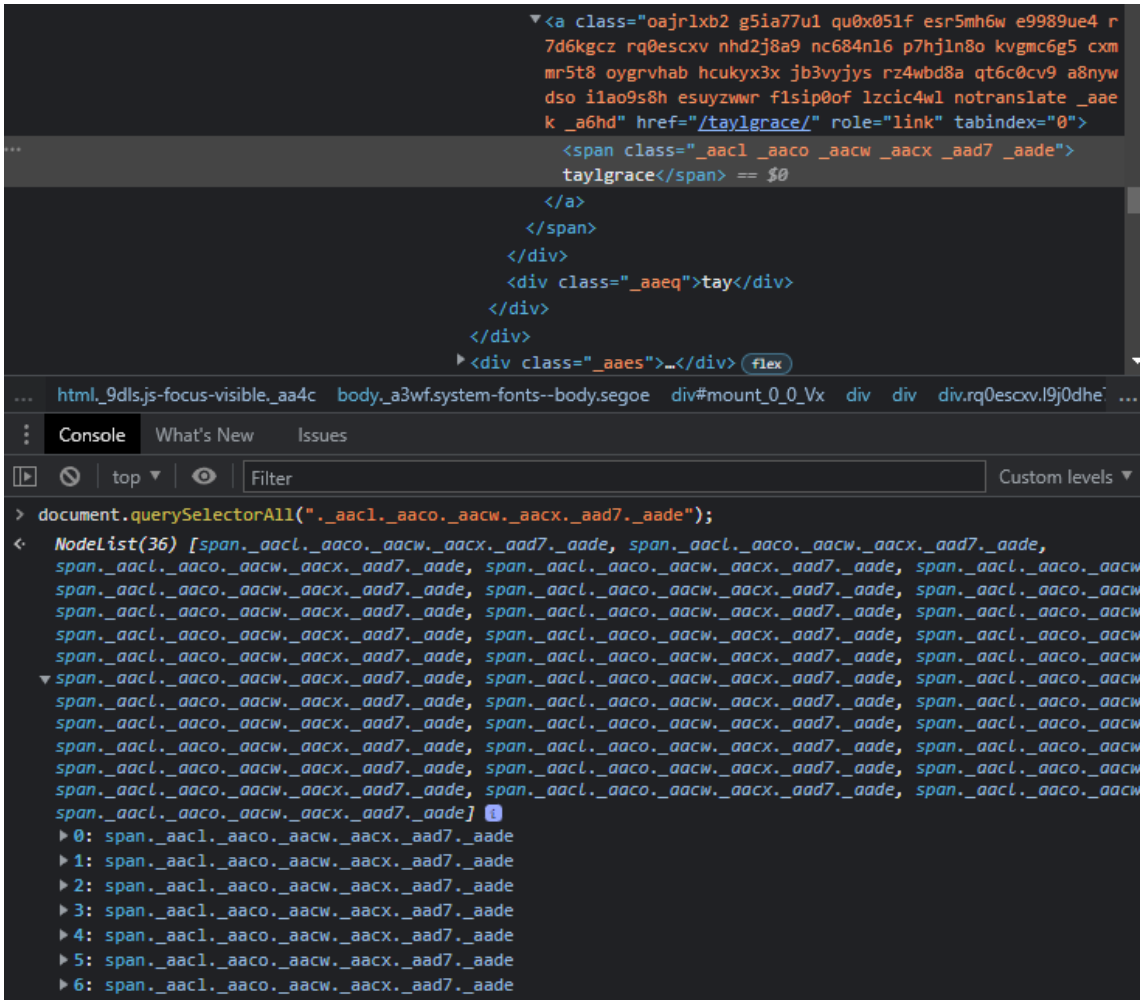

Figura 4.6: Exemplo de utilização da ferramenta "*DevTools*" para identificação dos marcadores utilizados pelo Instagram para a seção de seguidores

```
follower_elems = list(browser.find("._aacl._aaco._aacw._aacx._aad7._aade", waittime=1) or [])
followers_usernames: List[str] = list([ele.text for ele in follower_elems if ele.text != ""])
logging.info(f'Saving followers of \'{username}\'...({len(followers_usernames)})')
create_usernames_batch(followers_usernames)
```

Figura 4.7: Exemplo de utilização dos marcadores CSS para filtrar os elementos para a captura da lista de seguidores

4.4 Demonstração do Sistema

Com o intuito de demonstrar a utilização do sistema, essa seção descreve um ciclo típico de trabalho realizado pelo sistema. Para a realização da demonstração, foi utilizada uma conta do *Instagram* criada especificamente para este propósito.

O primeiro passo realizado consiste na adição de um *username* do *Instagram*

no banco de dados, que funcionará como ponto de partida para a realização dos trabalhos dos diferentes serviços, possibilitando a expansão do universo de domínio por meio da captura das listas de usuários que seguem ou são seguidos pelo perfil, além das interações realizadas por meio das postagens realizadas por este. Este passo pode ser realizado por adição direta no banco de dados, ou por meio da utilização das listas priorizadas e *watchlists* descritas na seção 4.2.1.

As figuras 4.8, 4.9 e 4.10 exemplificam as etapas de um ciclo de trabalho do serviço *instagram-crawler-description*. Na primeira é realizada uma chamada para a *EntityAPI* a fim de obter a lista de trabalho, que será utilizada para a captura de dados. Em seguida, é realizada a navegação no *Instagram* para o *username* apontado e efetuado a coleta dos dados, que são passados para a API, dando sequência à persistência das informações, como mostrado na figura 4.10.

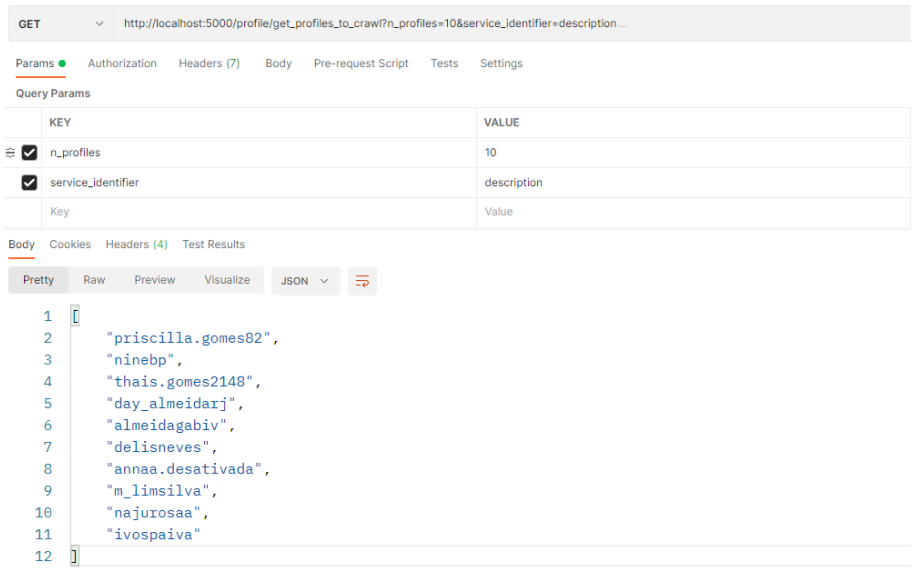

Figura 4.8: Exemplo de chamada do serviço *instagram-crawler-description* para obtenção da lista de perfis a crawlear.

À medida que os serviços povoadores (*instagram-crawler-description* e *instagram-crawler–follow*) são executados e o banco de dados populado, os serviços de contexto (*instagram-crawler–post*, *comment*, *instagram-crawler–likers*) podem ser executados, formando o cenário completo de um *username*, como pode ser verificado pela imagem 4.11, que ilustra a passagem das informações capturadas pelo *Post* para a *EntityAPI*.

Na figura 4.12 é demonstrada a realização de uma chamada à API para a extração

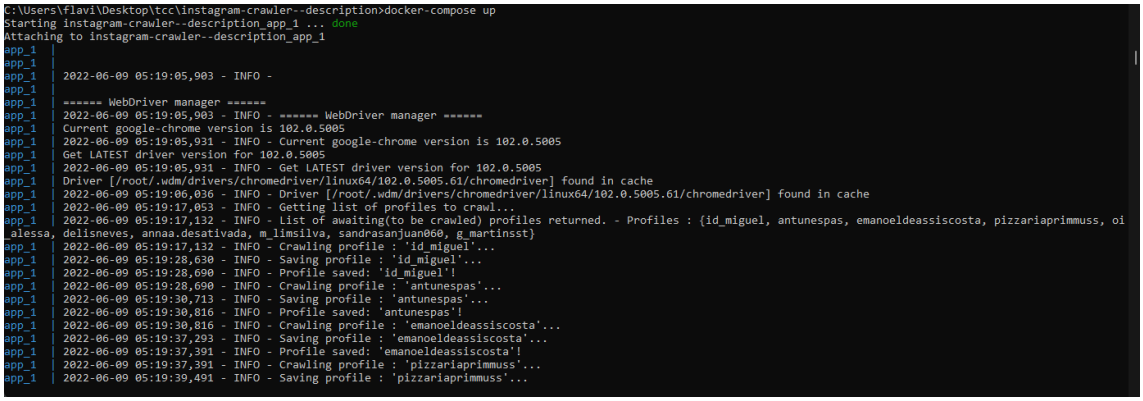

Figura 4.9: Exemplo de execução do serviço *instagram-crawler-description*

Figura 4.10: Exemplo de chamada do serviço *instagram-crawler-description* para a *EntityAPI*, persistindo o perfil coletado.

de informações presentes no banco de dados a respeito de um usuário do *Instagram*.

4.5 Resultados

Para os testes apresentados, o sistema foi executado numa máquina com a configuração: i3-10100F, 16GB de memória RAM, com Windows 11; no dia 03/07/22, pelo período de 1 hora. Os resultados obtidos foram dispostos na tabela 4.5, que relaciona a quantidade de usuários do Instagram identificados, a quantidade de postagens coletadas e a quantidade de downloads realizados. É importante ressaltar,

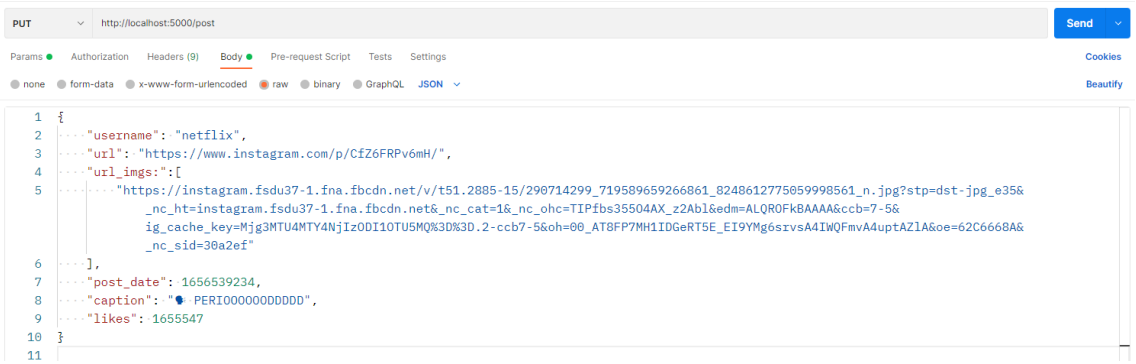

Figura 4.11: Exemplo de chamada do serviço *instagram-crawler-post* para a *EntityAPI*, persistindo o *post* coletado.

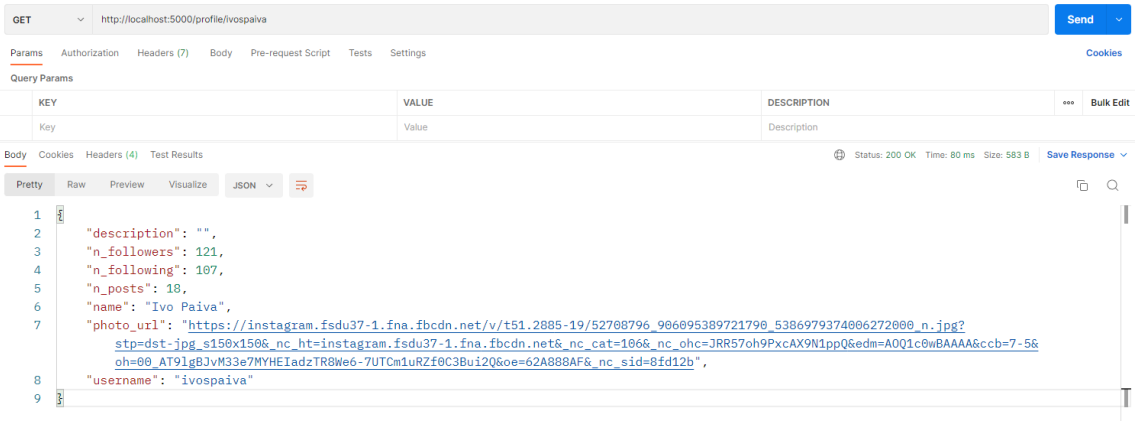

Figura 4.12: Exemplo de chamada para obter as informações de um perfil

que para os testes realizados, foram definidos limites para a quantidade de informações coletadas a partir de um *username* para cada serviço, sendo: 40 a quantidade máxima de *"scrolls"* realizados nos menus iterativos das seções *Follow* e *Following* de um profile e da seção *Likers* de um *Post*, e 20 a quantidade de postagens de um mesmo profile verificadas. Além disso, foi definido o intervalo de 10 segundos entre capturas de cada serviço.

Tais restrições foram empregadas para que o sistema utilizasse um ciclo de trabalho mais curto, possibilitando visitar mais itens dentro do intervalo de tempo definido, além de definir um fluxo menos intenso para tentar reduzir a probabilidade de sanções por parte do Instagram. Nesse sentido, ao longo do desenvolvimento do projeto, foi identificado a necessidade de realizar capturas mais intercaladas, visando emular o comportamento de um usuário típico da rede social.

Itens coletados	Quantidade
Usernames	7835
Posts	624
Mídias	624

Tabela 4.1: Estatísticas dos dados coletados ao longo dos testes, com a execução do projeto por 1 hora.

Capítulo 5

Conclusão

5.1 Considerações finais

No presente trabalho, foi apresentado um sistema dividido em cinco serviços[1][2][3][4][5] e uma API de integração[6], possibilitando o instanciamento de acordo com as necessidades do usuário. A aplicação se mostrou capaz de realizar a captura de dados a partir de uma conta do Instagram de maneira modular, podendo ser utilizada por outros trabalhos para a observação e aprimoramento de pontos específicos de captura.

Ao longo do trabalho, foram exemplificadas situações de uso das operações disponibilizadas pela aplicação proposta, além da identificação de possíveis estudos a partir de dados do Instagram.

A abordagem modular se mostrou eficaz para contornar o tempo necessário para a captura de informações mais lentas, tais como lista de seguidores e interações com o botão "gostei", que demandam navegação nos respectivos menus interativos, além de uma maior adaptabilidade frente às possíveis atualizações de interface por parte

[1] <https://gitlab.com/flaviowildner/instagram-crawler--description/-/tree/1.1.0>
[2] <https://gitlab.com/flaviowildner/instagram-crawler--follow/-/tree/1.1.0>
[3] <https://gitlab.com/flaviowildner/instagram-crawler--likers/-/tree/1.1.0>
[4] <https://gitlab.com/flaviowildner/instagram-crawler--post/-/tree/1.1.0>
[5] <https://gitlab.com/flaviowildner/instagram-crawler--post-downloader/-/tree/1.1.0>
[6] <https://gitlab.com/flaviowildner/instagram-crawler-entity-api/-/tree/1.1.0>

da rede social.

5.2 Limitações e trabalhos futuros

Apesar de útil para a captura de dados do Instagram, a ferramenta não possui, atualmente, a capacidade de extrair *Stories* e nem de realizar a captura a partir de *hashtags*. Além disso, a ferramenta proposta apresenta algumas limitações em função da sua forma de operação e da necessidade de manutenção frente às possíveis alterações realizadas na interface da rede social ao longo do tempo, uma vez que a captura de dados por meio de um *scraper* é realizada por meio da navegação na plataforma, tal como um usuário comum, acarretando em um tempo de operação maior, se comparado com as capturas realizadas por meio de APIs fornecidas por fontes proprietárias dos dados, por conta do *overhead* gerado pelas interfaces de apresentação que nada acrescentam em relação aos dados de interesse.

Em adição aos pontos mencionados, é importante frisar que a utilização de *scrapers* é tido como violação dos termos de uso na maioria das redes sociais, podendo acarretar em punições por parte das plataformas, que podem variar desde banimentos temporários de utilização até a exclusão do conta utilizada.

Como trabalho futuro, seria interessante o desenvolvimento de uma interface gráfica que disponibilizasse a possibilidade de controlar os diferentes serviços presentes no sistema, permitindo sua instanciação e a gestão do escalonamento, além de uma visualização dos dados ao longo do tempo e das estatísticas da captura em tempo real. Outro ponto a ser trabalhado é referente à forma de comunicação utilizada, sendo interessante realizar a comparação do padrão REST, aplicado no trabalho, com o modelo de utilização de filas, que pode apresentar maior aderência à arquitetura escalável, além de gerar um ganho de robustez contra possíveis falhas de comunicações temporárias entre os serviços e o escalonador de trabalho.

Referências

ACKOFF, R. L. From data to wisdom. *Journal of applied systems analysis*, v. 16, n. 1, p. 3–9, 1989.

ARAGÃO, F. B. P. et al. Curtiu, comentou, comprou. a mídia social digital instagram e o consumo. *Revista Ciências Administrativas*, Universidade de Fortaleza, v. 22, n. 1, p. 130–161, 2016.

AWAD, E.; GHAZIRI, H. *Knowledge Management*. [S.l.]: Pearson Education International, Upper Saddle River, NJ, 2004.

BASKARADA, S.; KORONIOS, A. Data, information, knowledge, wisdom (dikw): A semiotic theoretical and empirical exploration of the hierarchy and its quality dimension. *Australasian Journal of Information Systems*, v. 18, n. 1, 2013.

BOCJI, P. et al. *Business information systems: technology, development and management for the e-business*. 2nd. ed. [S.l.]: Financial Times/Prentice Hall, 2003.

BODDY, D.; BOONSTRA, A.; KENNEDY, G. *Managing Information Systems: an Organizational Perspective*. [S.l.]: Financial Times/Prentice Hall, 2005. ISBN 9780273686354.

CAMILO, C. O.; SILVA, J. C. d. Mineração de dados: Conceitos, tarefas, métodos e ferramentas. *Universidade Federal de Goiás (UFC)*, v. 1, n. 1, p. 1–29, 2009.

CARTA, S. et al. Popularity prediction of instagram posts. *Information*, v. 11, n. 9, 2020. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/11/9/453>.

CHAFFEY, D.; WOOD, S. *Business Information Management: Improving Performance using Information Systems*. [S.l.]: FT Prentice Hall, Harlow, 2005.

CORRÊA, Â.; SFERRA, H. Conceitos e aplicações de data mining. *Revista de ciência & tecnologia*, v. 11, n. 19-34, p. 20, 2003.

COUGHLIN, T. 175 zettabytes by 2025. *Forbes*, 2018. Disponível em: <https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/>.

DIXON, S. Leading countries based on instagram audience size as of january 2022. *Statista*, 2022. Disponível em: <https://www.statista.com/statistics/578364/countries-with-most-instagram-users/>.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37, 1996.

GOLDSCHMIDT, R.; PASSOS, E. *Data Mining: Um Guia Prático Conceitos*. [S.l.]: Elsevier, 2005.

GROFF, T.; JONES, T. *Introduction to Knowledge Management: KM in Business*. [S.l.]: Butterworth-Heinemann, 2003. ISBN 0750677287,9780750677288.

HIMAWAN, A.; PRIADANA, A.; MURDIYANTO, A. Implementation of web scraping to build a web-based instagram account data downloader application. *IJID (International Journal on Informatics for Development)*, v. 9, p. 59–65, 12 2020.

JAMES, J. What 'data never sleeps 9.0' proves about the pandemic. 2021. Disponível em: <https://www.domo.com/blog/what-data-never-sleeps-9-0-proves-about-the-pandemic/>.

JESSUP, L. M.; VALACICH, J. S. Information systems today. In: _____. [S.l.]: Upper Saddle River, N.J. : Prentice Hall, 2003.

KHDER, M. Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing and its Applications*, v. 13, p. 145–168, 12 2021.

LAUDON, K. C.; LAUDON, J. P. *Management information systems: Managing the digital firm*. [S.l.]: Pearson Educación, 2004.

MARQUESONE, R. *Big Data: Técnicas e tecnologias para extração de valor dos dados*. Casa do Código, 2016. ISBN 9788555192326. Disponível em: <https://books.google.com.br/books?id=cbWlDQAAQBAJ>.

PEARLSON, K. E.; SAUNDERS, C. S. *Managing and using Information Systems: a Strategic Approach*. [S.l.]: Wiley, New York, 2004.

REECE, A. G.; DANFORTH, C. M. Instagram photos reveal predictive markers of depression. *EPJ Data Science*, Springer, v. 6, n. 1, p. 15, 2017.

ROBILLARD, M. P. et al. Automated api property inference techniques. *IEEE Transactions on Software Engineering*, IEEE, v. 39, n. 5, p. 613–637, 2012.

ROWLEY, J. The wisdom hierarchy: representations of the dikw hierarchy. *Journal of information science*, Sage Publications Sage CA: Thousand Oaks, CA, v. 33, n. 2, p. 163–180, 2007.

SAURKAR, A. V.; PATHARE, K. G.; GODE, S. A. An overview on web scraping techniques and tools. In: . [S.l.: s.n.], 2018.

SMITH, K. 50 incredible instagram statistics. 2019. Disponível em: <https://www.brandwatch.com/blog/instagram-stats/>.

TAURION, C. Você realmente sabe o que é big data. *IBM Developer Works*, 2012.

TURBAN, E.; RAINER, R. K.; POTTER, R. E. *Introduction to information technology*. [S.l.]: John Wiley & Sons New York, NY, 2005.

ZAREI, K.; FARAHBAKHSH, R.; CRESPI, N. Deep dive on politician impersonating accounts in social media. In: *2019 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.: s.n.], 2019. p. 1–6.

ZHAO, B. Web scraping. In: _____. [S.l.: s.n.], 2017. p. 1–3. ISBN 978-3-319-32001-4.

ZIBRAN, M. F.; EISHITA, F. Z.; ROY, C. K. Useful, but usable? factors affecting the usability of apis. In: *2011 18th Working Conference on Reverse Engineering*. [S.l.: s.n.], 2011. p. 151–155.