Ricardo César Gonçalves Sant'Ana
Moisés Lima Dutra
Guilherme Ataíde Dias
Organizadores

ANAIS
WIDaT 2018

Organização do WIDaT 2018

- **Organização Geral:**

  Guilherme Ataíde Dias (PPGCI-UFPB) - Coordenador geral do evento

  Moisés Lima Dutra (PPGCIN-UFSC) - Vice-coordenador

- **Coordenador da Comissão Científica:**

  Ricardo César Gonçalves Sant'Ana (PPGCI-UNESP)

- **Comissão científica**

  Adilson Luiz Pinto (PPGCIN-UFSC)

  Ana Alice Baptista (Universidade do Minho, Portugal)

  Ana Carolina Simionato (PPGCI-UFSCar)

  Angela Maria Grossi de Carvalho (PPGCI-UNESP)

  Bernardina Maria Juvenal Freire de Oliveira (PPGCI-UFPB)

  Cristian Berrío-Zapata (PPGCI-UFPA)

  Dalton Lopes Martins (FCI-UnB)

  Denysson Axel Ribeiro Mota (PPGB-UFCA)

  Douglas Dyllon Jeronimo de Macedo (PPGCIN-UFSC)

  Ed Porto Bezerra (PPGI-UFPB)

  Edgar Bisset Alvarez (PPGCIN-UFSC)

  Edna Gusmão de Goés Brennand (MPGOA-UFPB)

  Edna Gomes Pinheiro (DCI-UFPB)

  Elaine Parra Affonso (FATEC-SP)

  Elvis Fusco (UNIVEM-Marília)

  Enrique Muriel Torrado (PPGCIN-UFSC)

  Evandro de Barros Costa (IC-UFAL)

  Fábio Paraguaçu (IC-UFAL)

  Fernando de Assis Rodrigues (PPGCI-UNESP)

  Gustavo Medeiros de Araújo (PPGCIN-UFSC)

  Henry Pôncio Cruz de Oliveira (PPGCI-UFPB)

  Joana Coeli Ribeiro Garcia (PPGCI-UFPB)

  José Eduardo Santarém Segundo (USP-FFCLRP)

  Leonardo Castro Botega (UNIVEM-Marília)

  Luana Farias Sales Marques (PPGCI-IBICT-UFRJ)

  Marckson Roberto Ferreira de Sousa (PPGCI-UFPB)

  Luís Fernando Sayão (CNEN)

  Marcelo Morandini (EACH-USP)

  Márcio Matias (PPGCIN-UFSC)

  Marcos Mucheroni (CBD-USP)

  Marynice de Medeiros Matos Autran (PPGCI-UFPB)

Maurício Barcellos Almeida (PPGGOC-UFMG)
Moisés Lima Dutra (PPGCIN-UFSC)
Plácida Leopoldina V. da Costa Santos (PPGCI-UNESP)
Pedro Luiz Pizzigatti Corrêa (POLI-USP)
Renata Baracho (PPGGOC-UFMG)
Ricardo César Gonçalves Sant'Ana (PPGCI-UNESP)
Robson Rodrigues Lemos (UFSC-Araranguá)
Rogério Ramalho (PPGCI-UFSCar)
Ryan Ribeiro de Azevedo (UFRPE-UAG)
Sandra de Albuquerque Siebra (PPGCI-UFPE)
Sandro Rautenberg (DECOMP-UNICENTRO)
Silvana Aparecida Borsetti G. Vidotti (PPGCI-UNESP)
Virginia Bentes Pinto (PPGCI-UFC)
Wagner Junqueira de Araújo (PPGCI-UFPB)
Zaira Regina Zafalon (PPGCI-UFSCar)

- **Coordenador do Cerimonial:**
  André Luiz Dias de França (PPGCI-UFPB)

- **Coordenador da Equipe Técnica Local:**
  Laerte Pereira da Silva Júnior (CCHLA-UFPB)

- **Equipe Técnica Local:**
  Adriana Alves Rodrigues (PPGCI-UFPB)
  Antonio Felipe dos Santos (MPGOA-UFPB)
  Débora Gomes de Araújo (PPGCI-UFPB)
  Pedro Augusto de Lima Barroso (PPGCI-UFPB)
  Pollianna Marys de Souza e Silva (PPGCI-UFPB)
  Renata Lemos dos Anjos (PPGCI-UFPB)

WEB SCRAPING DO RESEARCHERID:
proposta de sistema para o monitoramento de Índice H de pesquisadores no Brasil

WEB SCRAPING IN RESEARCHERID:
proposal to the monitoring system of H-index of the researchers in Brazil

Alexandre Ribas Semeler[1e2], Adilson Luiz Pinto[(2)] , Arthur Oliveira [(3)]
(1) Universidade Federal do Rio Grande do Sul, Instituto de Geociências, Campus do Vale Av. Bento Gonçalves, 9500 - Porto Alegre - RS - Brasil, alexandre.semeler@ufrgs.br.
(2) Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Ciência da Informação, Campus Professor João David Ferreira Lima - Trindade - Florianópolis - Santa Catarina - Brasil - CEP 88.040-900, adilson.pinto@ufsc.br.
(3) Universidade Federal do Rio Grande do Sul, Instituto de Informática, Campus do Vale Av. Bento Gonçalves, 9500 - Porto Alegre - RS - Brasil, arthur.holiver@gmail.com.

Resumo:
A dimensão dos dados aponta uma nova tendência de estudos e práticas que vem sendo adotada por cientistas da informação interessados em estudos métricos que visem o uso de dados de citação e referências. Essas abordagens ressaltam métodos e tecnologias que garantem a interoperabilidade e a criação de uma identidade única para autores e de documentos científicos. O ResearcherID é um sistema de identificação de autores científicos, criado em 2008 pela Thomson Reuters com o objetivo de resolver o problema da identificação de autores. O ResearcherID foi adotado pelo Conselho Nacional de Pesquisa do Brasil como fonte de dados para coleta do Índice H nacional. Este indicador é aplicado para medir a produtividade e para a visualização do impacto de cientistas baseando-se nos seus artigos mais citados. Nesse contexto, a proposta desse estudo será a de medir o índice H dos pesquisadores brasileiros cadastrados no ResearcherID. Os procedimentos metodológicos exigem a aplicação de conhecimentos inerentes a linguagem de programação *Python*. São utilizadas técnicas de web scraping e navegação web automatizada para recuperação de informação no ResearcherID. O resultado do estudo consolida-se na forma de dois *scripts* em Python (Apêndice A e Apêndice B) tais scripts objetivam-se como base para a elaboração de um sistema de monitoramento do índice H no ResearcherID.
Palavras-chave: Dados de Citação. ResearcherID. Índice H. Web Scraping.

Abstract:
The data dimension presents a new trends in studies and practices adopted by information scientists interested in metric studies that aim at the use of data citation and data references. These trends emphasize methods and technologies that ensure interoperability and the creation of a unique identity to authors and scientific documents. The ResearcherID is a system of identification of scientific authors, created in 2008 by Thomson Reuters with the purpose of solving the problem of the identification of authors. The ResearcherID was adopted by the National Research Council of Brazil as data source to validation and collection of the national H-index. This indicator is applied to measure the productivity and visualization of the impact of scientists based on their most cited articles. The aim of this study will be to measure the H-index of Brazilian researchers enrolled in ResearcherID. The methodological procedures require the application of knowledge inherent in the Python programming language. The web scraping and web surfing techniques are used for information retrieval in ResearcherID. The results of the study are consolidated in two Python scripts (Annex A and Annex B). These scripts are intended as a basis for the development of an H-index monitoring system in ResearcherID.
Keywords: Data Citation. ResearcherID. H-index. Web Scraping.

1 INTRODUÇÃO

A dimensão dos dados digitais aponta uma nova tendência de estudos e práticas que vem sendo adotada por cientistas da informação interessados em estudos métricos, em especifico aqueles que visem o uso de dados de citação bibliográfica.

Os estudos métricos com base neste tipo de dados não são algo novo, desde a década de 1963 são desenvolvidos pelo Institute for Scientific Information vide o exemplo do Science Citation Index, uma das bases de dados de citação científica mais importantes do mundo. A identificação precisa de autores e suas citações permite visualizar o seu impacto da produção científica. Isso também permite aos cientistas integrar suas publicações criar uma única identidade científica.

Neste contexto, propomos um estudo que irá utilizar a base de dados de citação do ResearacherID para identificar a variação do Índice H Brasileiro. Existem diversas outras fontes de dados de citação entre outras que fornecem o Índice H, Google Scholar, Scopus, ResearchGate entre outras. No entanto, ressalta-se que o Conselho Nacional de Pesquisas do Brasil utiliza a plataforma do ResearcherID para visualizar o número de citações e o Índice H dos pesquisadores Brasileiros.

O (CNPq) adotou o ResearcherID para validar os dados sobre citações a publicações científicas de impacto internacional na sua plataforma de currículos Lattes, vide Apêndice F. O CNPq utiliza o ResearcherID, para coletar e validar automaticamente os dados sobre o número total de citações, número de trabalhos e Índice H, dos pesquisadores que possuem publicações científica de impacto internacional na Web of Science por meio do ResearcherID.

Segundo Hirsch (2005), o Índice H é um indicador de quantificação para a produtividade e visualização do impacto de cientistas baseando-se nos seus artigos mais citados.

O resultado de busca por informação em dados de citação, principalmente, quando o foco está Índice H, pode indicar uma frente de pesquisa. Uma frente de pesquisa é composta pelo conjunto de autores mais citados em um determinado campo e indica a elite de autores de um campo científico e ou um pais. Nesse contexto, os dados de citação podem refletir a produtividade e a visibilidade dos autores em diversos campos científicos.

Tendo em vista o desenvolvimento das bases de dados de citação, conforme explanado por Rice e Southall (2016), a Clarivate Analitics desenvolveu em 2008 ResearcherID sistema utilizado para rastrear citações em conjuntos de dados da literatura científica de impacto internacional.

O ResearcherID é um sistema de identificação de autores científicos, criado pela Thompson Reuters com o objetivo de resolver o problema da identificação de autores. O ResearcherID se integra com a Web of Science (Science Citation Index), é um identificador digital único, gratuito e persistente, que distingue um pesquisador de outro e resolve o problema da ambiguidade e semelhança de nomes de autores e indivíduos, substituindo as variações de nome por um único código numérico facilitando o registro de informações ele serve para automatizar a atualização das publicações e de seus dados de citação possibilitando a interoperabilidade entre as publicações e suas citações a partir de uma única conta. (RESEARCHERID, 2018)

Esse ID científico é formado por um código de 11 dígitos, ex.: (J-9183-2016), na forma de caracteres alfanuméricos, sendo único para cada pesquisador cadastrado. O ResearcherID impede ambiguidades na identificação de autores e colaboradores em publicações. Seu funcionamento é

semelhante ao do Digital Object Identifier (DOI), código para objetos como artigos científicos, teses e dissertações (RESEARCHERID, 2018; RICE; SOUTHALL, 2016).

O ResearcherID é um identificador único que permite que os pesquisadores gerenciem suas listas de publicações, rastreiem suas contagens de citações e Índice h. (RESEARCHERID; 2018)

A utilização desses IDs favorece a localização e a contagem das publicações, além de reunir os dados de citações recebidas pelo conjunto de trabalhos de sua autoria possibilitando a visualização do Índice H.

Nesse sentido, a proposta desse estudo será a de medir o Índice H dos pesquisadores brasileiros cadastrados no ResearcherID, contar o número de publicações e visualizar o número de publicações com citação desses pesquisadores.

A seguir apresentam-se os objetivos, os procedimentos metodológicos e os resultados desse estudo com maiores detalhes.

2 OBJETIVOS

O objetivo geral desta proposta é desenvolver scripts em Python para monitorar o Índice H de pesquisadores brasileiros no ResearcherID. Para isso, propõem-se os seguintes objetivos específicos:

a) automatizar a coleta do Índice H de pesquisadores registrados no ResearcherID;
b) criar *scripts* de web scraping e navegação automatizada em linguagem de programação Python para coleta de dados de citação;
c) visualizar o Índice H dos pesquisadores brasileiros cadastrados no ResearcherID, contar o número de publicações e visualizar o número de suas publicações com citação.

3 PROCEDIMENTOS METODOLÓGICOS

O principal método adotado para este estudo foi o *web scraping*, método que se define como processo de extração e combinação de conteúdos web de forma sistemática e automatizada. Em tal processo, um agente de *software*, o *scraping*, simula o comportamento de navegação humano em servidores web copiando e reorganizando dados desorganizados em dados organizados (GLEZ-PENA et al., 2013).

Existem diferentes técnicas de web *scraping:* a manipulação de HTTP, a mineração de dados, as ferramentas *scraping*, a cópia manual e os microformatos. A manipulação de HTTP permite a colheita de dados estáticos e dinâmicos de um *site* através de uma solicitação HTTP. A mineração de dados é um processo automático que reconhece as informações de um site de acordo com *scripts* predefinidos que contêm dados incorporados. Ferramentas de *scraping* são *softwares* utilizados para extrair informações relacionadas a sites ou a funcionalidades e/ou estruturas de dados na *web*. As ferramentas de *scraping* também servem para extrair dados de redes sociais e são úteis para todos os tipos de atividades de *marketing* web. A cópia manual, embora existam ferramentas de *scraping* disponíveis, às vezes é necessária, por exemplo, em alguns casos em que a informação do site for bloqueada contra qualquer

forma web *scraping*. Por fim, os microformatos são referentes às tecnologias da web semântica, como dados abertos vinculados via RDF. Esses são conjuntos de informações geralmente elaboradas para serem intercambiadas via vocabulários e ontologias (WEBSTER, 2015).

O funcionamento de um web *scraping* ocorre da seguinte maneira: (a) acesso ao site via protocolo de comunicação web (HTTP), durante o processo de solicitação e resposta entre um cliente, normalmente um navegador da web, e um servidor da web; (b) o programa de *scraper* analisa o HTML e pode extrair seu conteúdo; (c) armazena o *output* em conteúdo textual, com o objetivo de transformar o conteúdo extraído em uma representação estruturada para posterior análise e armazenamento (GLEZ-PENA et al., 2013).

O objetivo da web *scraping* é transformar os dados desestruturados da *web* em representações estruturadas, como aquelas de formatos tabulares, por exemplo: os tipos de arquivos *Comma Separated Values* (CSV), *Tab Separated Values* (TSV), e/ou e*Xtensible Markup Languag*e (XML). Esses formatos permitem a analise do conteúdo de sites da mesma forma que permitem salvar a navegação feita pelos browsers web, como o Chrome ou FireFox.

No que segue apresenta-se uma sumarização dos procedimentos metodológicos, conforme o Quadro 1.

Quadro 1 – Caracterização do método

CARACTERIZAÇÃO DO MÉTODO	
TIPO DE PESQUISA	Exploratória e Descritiva
ESTRATÉGIA	Uso de linguagens de programação para coletar dados de citação
NATUREZA	Quantitativa
PANORAMA GERAL DA INVESTIGAÇÃO	Índice H de pesquisadores brasileiros
AMOSTRA	43.114 - IDS em 17 set. 2018
CORPUS TOTAL	43.114 - IDS em 17 set. 2018
INSTRUMENTOS DE COLETA	*Web scraping* e navegação web automatizada
SOFTWARES	*IDE* = *Pycharm* *Python*=2.7 *Módulos* = Codecs, BeautifulSoup, Selenium
FONTE DE COLETA DE DADOS	ResearcherID

Fonte: Dados da Pesquisa (2018).

Com base nesses procedimentos apresentam-se os resultados da pesquisa.

4 RESULTADOS

Os resultados obtidos nesse estudo representam um possível modelo para execução automatizada do processo de coleta de dados de citação oferecidos pelo ResearcherID. Os resultados subdividem-se em um plano técnico, criação de *scripts* para web *scraping* e navegação web automática

das informações no ResearcherID e em um plano metodológico de descrição e interpretação dos dados coletados sobre o Índice H brasileiro.

4.1 Resultados técnicos

Como resultado técnico obteve-se dois *scripts* em Python 2.7.15 os quais são aplicados para o monitoramento do Índice H dos pesquisadores brasileiros cadastrados no ResearcherID, contar o número de publicações e visualizar o número de publicações com citação desses pesquisadores com base no núcleo da Web of Science. Os mesmos podem ser visualizados no apêndice A e no apêndice B deste trabalho.

O apêndice A, apresenta o *script1*, o qual executa a primeira coleta de dados no ResearcheriD. O *script1* automatiza o web *scraping* das informações sobre: o nome de pesquisadores, sua instituição, o código ID e as áreas do conhecimento relacionadas ao seu ID. O *script1* salva uma lista contendo, o um conjunto de dados com todos os IDS de pesquisadores brasileiros que possuem registro no ResearcherID, a lista contém (43.114) pesquisadores brasileiros registrados em 19 de novembro de 2018. O script1 possui um dispositivo de tratamento de falhas, que garante que quando interrompida a extração, em caso de *timeout* na página, a extração seja reiniciada no último ponto valido coletado no Researcher. O script1 também evita as duplicações de IDs. O script1 gera um arquivo de log que contém apenas o último índice a ser extraído da lista usada. No começo da execução o script1 testa se já existe um arquivo com um índice, se o arquivo estiver vazio/não existir o índice a ser usado será 1 por padrão. Ao final da execução, caso o programa não seja interrompido por erro ou parada intencional do usuário, haverá no log de execução a data e hora em que a extração foi finalizada.

Os resultados da coleta feita pelo script1 são salvos em um arquivo *.TSV* . Está coleta e contém as seguintes informações: Nome do pesquisador, a instituição, o código no ResearcherID e as palavras chave. Estes dados servem para o *script2* como base para uma nova coleta de dados sobre as métricas de citação do *ResearcherID*.

O apêndice B representa o *script2,* o qual lê a lista de dados escrita pelo *script1* e executa a coleta de informações sobre as métricas de citação do ResearcherID. *O script2* lê os registros do arquivo .TSV e gera links *(http://www.researcherid.com/rid/+ ID* do pesquisador) usando os IDs do arquivo .TSV. Estes Links são únicos e representam os dados de citação de cada pesquisador. O *script2* possui o mesmo mecanismo de falhas do *script1*. O script2 é paralelizado em 8 processos iguais, cada processo coleta informações de (1/8) é feito de maneira aleatória na lista de links gerados a partir do do arquivo .TSV gerado pelo *script1*. As informações coletadas pelo script2 snao salvas em 8 arquivos .TSV diferentes durante a extração para acelerar o processo de coleta, no final da coleta as informações desses 8 arquivos são salvas em um único arquivo .TSV contendo os dados de citação da base, em especial: ID, total de publicações na lista, publicações com dados de citação, a soma de citações no tempo, a média de citações por artigo e o Índice H dos pesquisadores brasileiros. Por fim, o *script2* soma as publicações com dados de citação e faz a média do Índice H.

4.2 Interpretações da coleta

Os resultados da coleta permitem não apenas a visualização do Índice H, mas também outras informações apresentadas pelo ResearcherID como: total de publicações na lista, publicações com dados de citação, a soma de citações no tempo, a média de citações por artigo e o Índice H.

A descrição e a intepretação dos dados coletados permitiu visualizar Índices H discrepantes e aqueles que não correspondem a autores únicos, como aqueles utilizados para medir instituições ou mesmo aqueles feitos erroneamente. Estes dados foram excluídos da analise por não representarem o objetivo deste trabalho.

Assim, a coleta permitiu visualizar algumas situações, conforme o apêndice C, de um total de (40.114) pesquisadores cadastrados no ResearcherID, (9560) pesquisadores não possuem métricas no ResearcherID (N/A), ou seja nunca foram citados no Science Citation Index. Outros (960) possuem publicações com zero citação, ou seja, estes trabalhos não possuem impacto na WOS ou ainda não foram citados. As (10) maiores frequências de índice a H estão entre 1 e 10: (1=4006, 2=3454,3=3244, 4=2752, 5=2572, 6=2141, 7=1823, 8=1651, 9=1410, 10=1231)

Com base nos (40.114) IDs a média do Índice H nacional é (7,97), levando-se em consideração pesquisadores únicos e aqueles que possuem índice H maior que zero, ou seja, um montante de (32.255) pesquisadores brasileiros.

A frente de pesquisa nacional pode ser visualizada no apêndice D e concentra-se no campo cientifico da Física**.** Conforme a Tabela 2 contida no apêndice C, o ID (B-2946-2012) possui o maior índice H nacional válido (96). Este índice é de um professor do Instituto de Física Gleb Wataghin da Universidade Estadual de Campinas (UNICAMP). O segundo maior índice H (92) também corresponde a área de Física, conforme o ID (L-6239-2016), pertence a um professor do Instituto de Física da Universidade de São Paulo. Ambos atuam na área de Física Nuclear e são Bolsistas de Produtividade em Pesquisa do CNPq - Nível 1D.

O apêndice D também permite visualizar e ranquear os pesquisadores brasileiros conforme a variação dos Índices H mais altos (mínimo 75 e máximo 96) são únicos. O ID que recebeu mais citações (39544) é (L-1621-2016) de um professor da Universidade Estadual do Rio de Janeiro que também atua na área de Física nuclear sendo Bolsista de Produtividade em Pesquisa do CNPq - Nível 1C.

Já as publicações com citação na WOS compreendem o montante de (877.624) estas são publicações que receberam no mínimo 1 citação em periódico com fator de impacto. O número total de publicações registradas nas listas foi (1.146.855), no entanto este número é prejudicado ao representar o número de produções do Brasil na WOS, pois alguns pesquisadores não preencheram seus IDs apenas com a sua produção na WOS mas com outras fontes de dados de citação como Google Scholar e Scopus.

Por fim, o ID (K-5168-2013) possui a maior média de citações recebidas (99.88) no ReseacherID a coordenadora do Núcleo de Pesquisa em Asma e Inflamação das Vias Aéreas (NUPAIVA), com índice H (22) a pesquisadora da Universidade Federal de Santa Catarina valida toda sua produção na Plataforma Lattes.

No entanto, chamou a atenção que os maiores índices H brasileiros (140), (105) e (103) não serem válidos, pois foram realizados com objetivos diversos e não com o objetivo de validar os

dados de citação no Lattes CNPq. É necessário ter em mente que o ReserchерID é um identificador pessoal de autor e possui o objetivo de fornecer métricas para produção de autores únicos e não de instituições, ressaltando também que não é uma ferramenta para pesquisa bibliográfica.

Assim o índice H 140, (A-9780-2017), atribuído ao Instituto de Física, Gleb Wataghin da UNICAMP e o índice H 105, (M-2664-2016), que representa o Instituto de Física de São Carlos da Universidade de São Paulo prejudicam a visualização do índice H nacional de autores já que são índices de um conjunto de autores de uma instituição. Outra discrepância é o índice H 103 (E-4724-2015), de um estudante do Laboratório de Raioisótopos do Centro de Energia Nuclear na Agricultura – CENA, que foi construído de maneira errônea pois o autor em sua lista pessoal de publicações insere produções científicas de outros autores de certo ele utilizou a plataforma para visualizar a produção em sua área de pesquisa desconhecendo o objetivo do ResearcherID, tão pouco ele valida seu Lattes com os dados.

Estes três casos ressaltam a relevância desta proposta de trabalho. Pois a visualização do Índice H nacional é usada pelo CNPq como estratégia para fomentar e desenvolver novas pesquisas, pautando-se na relevância e no impacto dos pesquisadores atribuída pela WOS, nas mais diversas áreas do conhecimento. No entanto, é necessário fazer o uso correto da plataforma ResearcherID que é um instrumento de validação de citações utilizada pela plataforma Lattes do CNPq.

4 CONSIDERAÇÕES FINAIS

O Índice H é usado pelo CNPq como um dos indicadores para pautar a distribuição de Bolsas de Pesquisa e para pautar a relevância acadêmica da produção cientifica no Brasil. Conhecer o Índice H nacional torna-se tarefa de suma relevância. Pois o Índice H indica o impacto e a relevância dos trabalhos mais citados ao longo do tempo. Revelando a visibilidade e o uso da produção científica de impacto internacional de cada pesquisador.

Considera-se esse estudo como uma estratégia fundamental para medir a variação do Índice H nacional, a variação da média de citações de impacto internacional, medir a soma das citações da produção científica nacional, assim como para visualizar o total de publicações nacionais nas listas do ResearcherID com citações.

Por fim, ressalta-se que os dados de citação e os IDs de autores são ferramentas necessária para visualização do impacto internacional do que é produzido em nosso país, sendo automatização desse processo de suma aplicabilidade em estudos métricos que visem o uso de dados para qualificar a produção cientifica nacional. Conclui-se que os usuários brasileiros deveriam ter maior critério ao utilizar o ResearcherId, pois a ferramenta é utilizada pelo CNPq para validar o impacto da produção nacional na Web of Science uma das principais e mais respeitadas bases de dados e documentos bibliográficos.

Agradecimentos

"O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001"

REFERÊNCIAS

HEIRICH, J. An index to quantify an individual's scientific research output. **PNAS, v.** 102, n. (46), 2005. Disponível em: https://doi.org/10.1073/pnas.0507655102>. Acesso em: set. 2018.

GLEZ-PEÑA, D. et al.. Web scraping technologies in an API world. **Briefings in Bioinformatics**, v. 15, n. 5, p. 788-797, 2013. Disponível em: <http://bib.oxfordjournals.org/content/15/5/788>. Acesso em: set. 2018.

ResearcherID. Disponível em: < http://www.researcherid.com > Acesso em: set. 2018.

RICE, R.; SOUTHALL, S. **The data librarian's handbook**. London: Facet Publishing, 2016.

WEBSTER, S. **What Is Scraping?** The Basics For Everyone. 2015. Disponível em: https://myhelpster.com/what-is-scraping-the-basics-for-everyone/. Acesso em: set. 2018.

PYTHON. Disponível em: <https://www.python.org/>. Acesso em set. 2018.

Apêndice A – ***Script1:*** coleta de informações gerais no ResearcherID

```
# coding: utf-8
"""
Created on Sep 17 2018
"""

import codecs
from bs4 import BeautifulSoup
import selenium.common.exceptions
# from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as ec
from selenium.webdriver import Firefox
# import os

url = 'http://www.researcherid.com/ViewProfileSearch.action'    # define url alvo
# Caso a execução seja feita com Chrome
# chromedriver = "../scripts/chromedriver"            # path do chromedriver
# os.environ["webdriver.chrome.driver"] = chromedriver
# driver = webdriver.Chrome(chromedriver)                         # abre browser

# abre arquivo para escrita dos resultados
with codecs.open('../dados/csv/RESEARCHID_Extraction2.tsv', 'w', 'utf-8-sig') as re-
sults:
    results.write('Name\tInstitution\tResearcher ID\tKeywords\n')   # escreve headers
results.close()
id_list = []
counter = 0     # inicializa contador de resultados e erros
errors = 0

driver = Firefox()
while True:
    try:
        driver.get(url)                                                 # navega para a
url
        # espera carregamento da combobox para escolher o Pais alvo
        WebDriverWait(driver, 10).until(ec.presence_of_element_located((By.CSS_SELEC-
TOR,
                                                                        '#country >
option:nth-child(33)'))).click()
        break
    except selenium.common.exceptions.TimeoutException:
        continue

# clica para submeter pesquisa
driver.find_element_by_css_selector('#submitImage > img').click()
# define limite de paginas para serem acessadas nos resultados
page_limit = int(WebDriverWait(driver, 10).until(ec.presence_of_element_located(
    (By.CSS_SELECTOR, '#criteria_requestedPage_total'))).text) // 50 + 1
# espera combobox carregar para mostrar 50 resultados por pagina
WebDriverWait(driver, 10).until(ec.presence_of_element_located((By.CSS_SELECTOR,
                                                               '#resultsPerPage >
option:nth-child(3)'))).click()

for page in range(page_limit):      # percorre paginas de resultados da busca
    # espera tabela de resultados ser carregada
    WebDriverWait(driver, 10).until(ec.presence_of_element_located(
        (By.CSS_SELECTOR, '#resultContainer > table:nth-child(23) > tbody > tr:nth-
-child(2) > td > table > tbody')))
    # define raiz da arvore de resultados para parsing
    root = BeautifulSoup(driver.page_source, 'lxml').find('td', {'class': 'resultsTa-
bleBox'}).findNext('tbody')
    with codecs.open('../dados/csv/RESEARCHID_Extraction2.tsv', 'a', 'utf-8-sig') as
results:
        for item in root.findAll('tr')[1:]:     # percorre itens na tabela
            data = item.findAll('td')           # lista todos os resultados
            rid = data[4].getText().replace('\n', '').replace(' ', '')
            if rid in id_list:
                continue
            else:
                id_list.append(rid)
                # escreve resultados no arquivo de output
```

```
                results.write('%s\t%s\t%s\t%s\n' % (data[1].a.getText().replace('\n',
'),
                                                    data[2].getText().replace('\n',
'),
                                                    rid,
                                                    data[5].getText().replace('\n',
')))
                counter += 1        # incrementa contador de resultados
        results.close()
    if page < page_limit:       # testa se as paginas de resultados ja acabaram
        # clica em next page caso contrario
        WebDriverWait(driver, 10).until(ec.presence_of_element_located(
            (By.CSS_SELECTOR, '#resultContainer > table:nth-child(22) > tbody > tr >
td:nth-child(2) > table > '
                              'tbody > tr > td:nth-child(7) > span > img'))).click()
    print counter
    try:        # tenta esperar pelo carregamento de um item da pagina como teste
        WebDriverWait(driver, 20).until(ec.presence_of_element_located(
            (By.ID, 'pageNumnull')))
    except selenium.common.exceptions.TimeoutException:     # em caso de timeout,
trata o erro
        driver.get(url)     # navega para a url alvo novamente
        errors += 1         # incrementa counter de erros
        # reinicia a pesquisa
        WebDriverWait(driver, 30).until(ec.presence_of_element_located(
            (By.CSS_SELECTOR, '#country > option:nth-child(33)'))).click()
        driver.find_element_by_css_selector('#submitImage > img').click()
        WebDriverWait(driver, 40).until(ec.presence_of_element_located(
            (By.CSS_SELECTOR, '#resultsPerPage > option:nth-child(3)'))).click()
        page -= 1       # reduz contador de paginas navegadas em 1
        element = WebDriverWait(driver, 10).until(ec.presence_of_element_located((By.
XPATH, '//*[@id="pageNumnull"]')))
        element.clear()     # limpa caixa de texto com numero de paginas
        element.send_keys('%s' % page)      # escreve numero de paginas na caixa de
texto
        driver.find_element_by_xpath('/html/body/div[3]/form[2]/table/tbody/tr/td/div/
table/tbody/tr[2]/td[2]/div/div/'
                                     'table[1]/tbody/tr/td[2]/table/tbody/tr/td[5]/a/
img').click()  # clica em "go"
driver.quit()
print errors, 'errors happened. Extraction finished.'
```

Fonte: Dados da Pesquisa (2018).

Apêndice B – Script2: coleta das métricas de citação do ResearcherID

```
# coding: utf-8
"""
Created on Aug 27 2018
"""
import codecs
from bs4 import BeautifulSoup
from multiprocessing import Pool
import selenium.common.exceptions
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as ec
from selenium.webdriver.firefox.options import Options
from selenium.webdriver import Firefox
# import os
# from selenium.webdriver.chrome.options import Options
# from selenium import webdriver


################################################################################################
###############
master = codecs.open('../dados/csv/RESEARCHERID-ALL-METRICS.tsv', 'r', 'utf-8')
id_list = master.readlines()[1:]
master.close()
for x in range(len(id_list)):
    id_list[x] = '%s' % id_list[x].split('\t')[0][1:]
################################################################################################
###############
options = Options()
options.add_argument("--headless")


class Control:
    def __init__(self):
        self.counter = 0
        self.article_accumulator = 0
        self.with_citation_accumulator = 0
        self.h_index_accumulator = 0
        self.existent_h_index_counter = 0


def id_cleaning(a_link):
    if '\r\n' in a_link:  # testa finais de linha para verificar se usam \n ou \r\n
        return a_link.replace(' ', '').replace('\r\n', '\t')  # troca final por \t e tira
blank spaces
    else:
        return a_link.replace(' ', '').replace('\n', '\t')


def get_metrics_info(a_soup):   # function que extrai dados alvo da pagina
    return (a_soup.find(id='metrics_totalArticleCount').getText(),
            a_soup.find(id='metrics_articleCountForMetrics').getText(),
            a_soup.find(id='metrics_timesCited').getText(),
            a_soup.find(id='metrics_averagePerItem').getText(),
            a_soup.find(id='metrics_hindex').getText())


def parallel_scraping(a_trio):
    global id_list
    driver = Firefox(firefox_options=options)
    # driver = webdriver.Chrome(chromedriver)
    for item in a_trio[0]:  # for para percorrer a lista de links
        rid = id_cleaning(item.split('\t')[2])
        if rid in id_list:
            continue
        link, name = 'http://www.researcherid.com/rid/' + rid, item.split('\t')[0]
        a_trio[2].counter += 1  # incrementa counter para acompanhar progresso
        print a_trio[2].counter, ': Extracting: %s' % link

        driver.get(link)  # acessa link alvo
        try:
```

```
            driver.find_elements_by_xpath('/html/body/div[3]/div/table/tbody/tr/td[1]/tab-
le/tbody/tr[1]/td[2]/'
                                          'div[1]/ul[1]/''li[2]/a')[0].click()
            # clica para mostrar citation metrics
        except IndexError:
            print '%s apparently got no citation metrics' % link  # exception tratada
            continue
        a_file = codecs.open(a_trio[1], 'a', 'utf-8-sig')
        try:
            WebDriverWait(driver, 10).until(ec.presence_of_element_located(
                (By.CSS_SELECTOR, '#metrics_hindex')))
        except selenium.common.exceptions.TimeoutException:
            a_file.write(link.split('/')[-1] + '\t' + 'TO\tTO\tTO\tTO\tTO\n')
        else:
            root = BeautifulSoup(driver.page_source, 'lxml')  # parsing inicial
            data = get_metrics_info(root)
            if data != '':
                a_trio[2].article_accumulator += int(data[0])
                a_file.write(link.split('/')[-1] + '\t' + '%s\t%s\t%s\t%s\t%s\n' % data)
                if data[-1] > 0:
                    a_trio[2].with_citation_accumulator += int(data[1])
                    a_trio[2].h_index_accumulator += float(data[-1])
                    a_trio[2].existent_h_index_counter += 1
            else:
                a_file.write(link.split('/')[-1] + '\t' + 'NA\tNA\tNA\tNA\tNA\n')
                # aguarda carregamento da pagina para escrever no arquivo de output
        finally:
            a_file.close()
    driver.quit()  # fecha browser


if __name__ == '__main__':
    pool = Pool(processes=8)
    # geckodriver = "../scripts/geckodriver"    # path do geckodriver para firefox

    # chrome_options = Options()
    # chrome_options.add_argument("--headless")
    # chromedriver = "../scripts/chromedriver"    # path do chromedriver
    # os.environ["webdriver.chrome.driver"] = chromedriver

    with codecs.open('../dados/csv/RESEARCHID_Extraction2.tsv', 'r', 'utf-8-sig') as re-
search:      # abre arquivo
        link_list = research.readlines()[1:]    # extrai linhas do arquivo em uma lista
    research.close()                            # fecha arquivo

    with open('../dados/error_logs/ResearcherIDanonScraping.txt', 'w') as log:
        out_file1 = codecs.open('../dados/csv/RESEARCHIDresultsPart1-8.tsv', 'w', 'utf-
-8-sig')  # abre arquivo de output
        out_file2 = codecs.open('../dados/csv/RESEARCHIDresultsPart2-8.tsv', 'w', 'utf-
-8-sig')  # abre arquivo de output
        out_file3 = codecs.open('../dados/csv/RESEARCHIDresultsPart3-8.tsv', 'w', 'utf-
-8-sig')  # abre arquivo de output
        out_file4 = codecs.open('../dados/csv/RESEARCHIDresultsPart4-8.tsv', 'w', 'utf-
-8-sig')  # abre arquivo de output
        out_file5 = codecs.open('../dados/csv/RESEARCHIDresultsPart5-8.tsv', 'w', 'utf-
-8-sig')  # abre arquivo de output
        out_file6 = codecs.open('../dados/csv/RESEARCHIDresultsPart6-8.tsv', 'w', 'utf-
-8-sig')  # abre arquivo de output
        out_file7 = codecs.open('../dados/csv/RESEARCHIDresultsPart7-8.tsv', 'w', 'utf-
-8-sig')  # abre arquivo de output
        out_file8 = codecs.open('../dados/csv/RESEARCHIDresultsPart8-8.tsv', 'w', 'utf-
-8-sig')  # abre arquivo de output
        file_list = [out_file1, out_file2, out_file3, out_file4, out_file5, out_file6, out_file7,
out_file8]
        for out_file in file_list:
            out_file.write('ResearcherID\tTotal in Pub. List\tWith Citation Data\tSum of
Times Cited\t'
                           'Average Citations/Article\tH-index\n')   # escreve headers
            out_file.close()
        control1, control2, control3, control4, control5, control6, control7, control8 =
Control(), Control(), \

Control(), Control(), \
```

```
    Control(), Control(), \

    Control(), Control()
            control_list = [control1, control2, control3, control4, control5, control6, con-
    trol7, control8]
            print 'Begin of extraction,', len(link_list) - 1, 'items to go'

            octan = len(link_list) // 8
            pool.map(parallel_scraping, [(link_list[:octan-1], '../dados/csv/RESEARCHIDresul-
    tsPart1-8.tsv', control1),
                                         (link_list[octan:2*octan-1], '../dados/csv/RESEARCHI-
    DresultsPart2-8.tsv', control2),
                                         (link_list[2*octan:3*octan-1], '../dados/csv/RESEAR-
    CHIDresultsPart3-8.tsv', control3),
                                         (link_list[3*octan:4*octan-1], '../dados/csv/RESEAR-
    CHIDresultsPart4-8.tsv', control4),
                                         (link_list[4*octan:5*octan-1], '../dados/csv/RESEAR-
    CHIDresultsPart5-8.tsv', control5),
                                         (link_list[5*octan:6*octan-1], '../dados/csv/RESEAR-
    CHIDresultsPart6-8.tsv', control6),
                                         (link_list[6*octan:7*octan-1], '../dados/csv/RESEAR-
    CHIDresultsPart7-8.tsv', control7),
                                         (link_list[7*octan:], '../dados/csv/RESEARCHIDresul-
    tsPart8-8.tsv', control8)])
        total_H_index = 0
        total_existent_H_index = 0
        total_cited = 0
        total_articles = 0
        elements_counter = 0
        for x in control_list:
            total_H_index += x.h_index_accumulator
            total_existent_H_index += x.existent_h_index_counter
            total_cited += x.with_citation_accumulator
            total_articles += x.article_accumulator
            elements_counter += x.counter

        log.write('Total count: %s\n' % str(elements_counter))
        log.write('Average H-Index: %s\n' % str(total_H_index / total_existent_H_index))
        log.write('Total articles with citation data: %s\n' % str(total_cited))
        log.write('Total articles: %s\n' % str(total_articles))
        log.close()
        print 'Extraction finished:', elements_counter, 'elements listed'
```

Fonte: Dados da Pesquisa (2018).

Apêndice C – Tabela 1 – Rank de Frequências do Índice H Brasileiro

H index	Freq.	H index	Freq.
0	960	41	17
1	4006	42	25
2	3454	43	5
3	3244	44	21
4	2752	45	5
5	2572	46	12
6	2141	47	10
7	1823	48	12
8	1651	49	16
9	1410	50	7
10	1231	51	6
11	1082	52	3
12	931	53	9
13	748	54	9
14	671	55	7
15	604	56	7
16	526	57	1
17	498	58	7
18	369	59	5
19	355	60	4
20	307	61	3
21	253	62	6
22	251	63	1
23	205	67	3
24	183	68	2
25	159	69	5
26	126	71	2
27	143	73	1
28	96	75	3
29	85	76	3
30	90	77	2
31	66	80	2
32	46	83	2
33	57	84	2
34	37	85	1
35	42	86	1
36	38	92	1
37	22	96	1
38	26	103	1
39	15	105	1
40	28	140	1
		N/A	9576

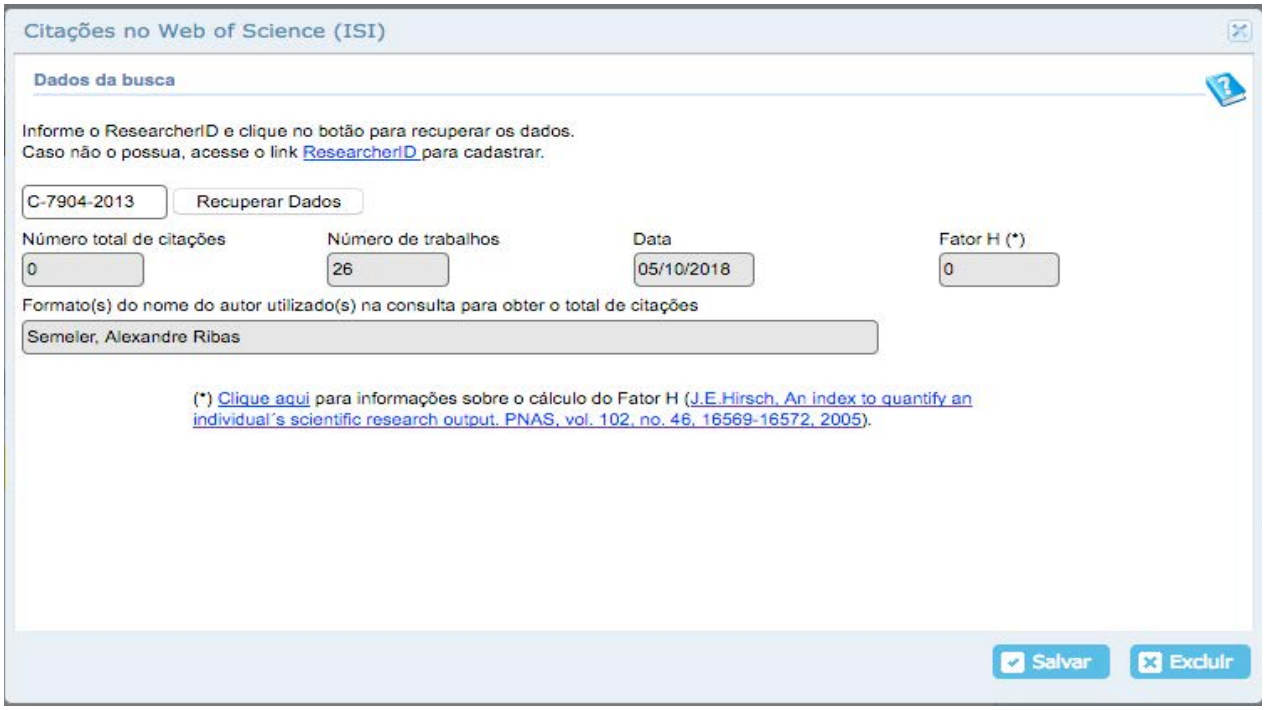
Fonte: Dados da Pesquisa (2018).

Apêndice D – Frente de Pesquisa Nacional Conforme Índice H

Rank/ ID		Total in Pub. List	With Cita-tion Data	Sum of Times Cited	Average Citations/ Article	H-index	Área
1	B-2946-2012	438	438	31.845	72.87	96	Física (UNI-CAMP)
2	L-6239-2016	408	406	29.667	73.07	92	Física (USP)
3	D-3532-2012	833	820	39.622	48.50	86	Física (UNESP)
4	L-1621-2016	866	745	39.544	53.22	85	Física (UERJ)
5	C-4007-2013	619	619	37.172	60.15	84	Física (UFBA)
6	D-4476-2013	520	520	27.898	53.65	84	Medicina (UFPEL)
7	C-7679-2016	611	298	34.566	115.99	83	Medicina (UFRJ)
8	G-9573-2012	755	755	25.892	34.29	80	Psicologia (UFRGS)
9	L-4142-2016	501	488	28.821	59.30	77	Física (UERJ)
10	E-8874-2010	459	321	18.207	56.72	75	Física (USP)

Fonte: Dados da Pesquisa (2018).

Apêndice F – Validação do Índice H no Lattes CNPq



Fonte: Dados da Pesquisa (2018).