UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS SOCIAIS APLICADAS
DEPARTAMENTO DE RELAÇÕES INTERNACIONAIS
BACHARELADO EM RELAÇÕES INTERNACIONAIS

ROMBERG DE SÁ GONDIM

SOLUÇÕES PARA COLETA E TRATAMENTO DE DADOS QUALITATIVOS SOBRE DIFUSÃO E APROPRIAÇÃO DA AGENDA 2030: A LINGUAGEM PYTHON APLICADA À CONSTRUÇÃO DE UM CORPUS TEXTUAL

João Pessoa
2023

ROMBERG DE SÁ GONDIM

SOLUÇÕES PARA COLETA E TRATAMENTO DE DADOS QUALITATIVOS SOBRE DIFUSÃO E APROPRIAÇÃO DA AGENDA 2030: A LINGUAGEM PYTHON APLICADA À CONSTRUÇÃO DE UM CORPUS TEXTUAL

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Relações Internacionais pela Universidade Federal da Paraíba.

Orientador: Dr. Pascoal Teófilo Carvalho Gonçalves

JOÃO PESSOA

2023

Catalogação na publicação
Seção de Catalogação e Classificação

G637s Gondim, Romberg de Sá.
Soluções para coleta e tratamento de dados qualitativos sobre difusão e apropriação da Agenda 2030: a linguagem Python aplicada à construção de um corpus textual / Romberg de Sá Gondim. - João Pessoa, 2023.
37 f.

Orientação: Pascoal Teófilo Carvalho Gonçalves. TCC (Graduação) - UFPB/CCSA.

1. Pesquisa qualitativa. 2. Coleta de dados. 3. Tratamento de dados. 4. Análise de Conteúdo. 5. Discurso Político. 6. Rotinas de programação. I. Gonçalves, Pascoal Teófilo Carvalho. II. Título.

UFPB/CCSA CDU 327

Elaborado por ANA CLAUDIA LOPES DE ALMEIDA - CRB-15/108

ROMBERG DE SÁ GONDIM

SOLUÇÕES PARA COLETA E TRATAMENTO DE DADOS QUALITATIVOS SOBRE DIFUSÃO E APROPRIAÇÃO DA AGENDA 2030: A LINGUAGEM PYTHON APLICADA À CONSTRUÇÃO DE UM CORPUS TEXTUAL

Trabalho de Conclusão de Curdo apresentado ao Curso de Relações Internacionais do Centro de Ciências Sociais Aplicadas (CCSA) da Universidade Federal da Paraíba (UFPB), como requisito parcial para obtenção do grau de bacharel (a) em Relações Internacionais.

Aprovado(a) em, 09 de março de 2023

BANCA EXAMINADORA

Prof. Dr. Pascoal Teófilo Carvalho Gonçalves – (Orientador)
Universidade Federal da Paraíba - UFPB

Prof. Dr. Túlio Sérgio Henriques Ferreira
Universidade Federal da Paraíba - UFPB

Prof. Dr. Thiago Lima da Silva
Universidade Federal da Paraíba - UFPB

A todos aqueles que errando, titubeando ou estranhando meu nome pela primeira vez, tiveram tempo para corrigir, ou brincar sobre nas muitas outras vezes que conversamos.

À minha mãe (in memoriam), e a todos os seus ensinamentos.

AGRADECIMENTOS

Ao longo da minha formação, era dizer corriqueiro do meu pai, professor, a frase de Newton “Se eu vi mais longe, foi por estar nos ombros de gigantes”. No entanto, gostaria aqui de trazer uma frase um tanto parecida, mas que expressa em maior grau, a meu ver, a dualidade do internacionalista. Proferida na solenidade de criação da Universidade Federal da Paraíba, em 1955, José Américo de Almeida disse: “Outros vos darão asas; eu vos dou as raízes”. Enxergo o internacionalista como um preso na liberdade entre as asas do internacional e as raízes saudosas do local.

No entanto, gostaria de iniciar ultrapassando essa nossa dualidade de conceitos tão distintos como interdependentes: de anarquia e hierarquia, de agência e estrutura, de local e internacional, e agradecer a todos aqueles que, na minha vida, foram e são tanto raízes como asas. Abaixo, listei alguns muitos.

À minha mãe, Lenilde Duarte de Sá (in memoriam), meu Farol do Cabo Branco, a quem prometi “ser bom, ser bom e ser bom”. Nos muitos encontros que ainda temos, por meio das coincidências acadêmicas e pessoais da vida, me alegro com a sua presença. Agradeço por me indicar o internacional, e espero seguir cumprindo com a promessa feita. Te amo mais que o céu e que as estrelas.

Ao meu pai, Romberg Rodrigues Gondim, meu Farol do Bacupari, que me ensinou a importância de trilhar a felicidade e o prazer onde estiver, com quem estiver. Obrigado por todas as lições, por todos os momentos em que você me acalmou e me orientou da sua forma. Obrigado pelas lições e histórias repetidas, mesmo que você já saiba que eu já sei. Obrigado pela sua energia sempre alegre, tranquila e jovem. Te amo muito, paizão, e não há nada que nos separe.

Aos meus pais em conjunto e à minha irmã, Cibelle Gondim, na condição de professores dentro e fora da sala de aula, e por despertar a minha paixão pela academia, pela pesquisa e pela docência. Obrigado por sempre me incentivarem e confiarem.

Ao meu amor, Victória, por cada segundo. Obrigado por me tranquilizar nos momentos mais carregados, e por me motivar nos momentos mais difíceis. Sou grato nada mais nada menos do que por você, em todos os seus lindos sorrisos e pelos milissegundos que antecedem cada abraço.

À minha família, em especial meus tios e “prirmões” de sangue e de coração. Tia Ana, Tia Zi e Tio Raglan, amo muito vocês e nossas farras. Tio Aldo, Tias Glória, Tânia, Aninha, Ana Coutinho, Celhinha, Tia Vera, obrigado por tratarem como gigantes cada pequena conquista, e perdão pelos sumiços. Aos meus primos queridos Lucas, Gabriel, Nina, Tati e Mari.

À praia do Seixas, minha primeira casa, e àqueles que ela me trouxe. Mamá, Duda e Poli, obrigado pela amizade eterna, pelas brincadeiras de criança que ainda tomam conta de nós ainda depois de muitos anos. Gostaria de deixar também um abraço especial à Genival, por todos os ensinamentos e histórias.

Aos meus amigos queridos Lucas Gondim, Pedro Guerra, Bolivar, Letícia Viana, Heitor, Ribeiro, Carol, Isadora, Duda e Ian, por todo o carinho e parceria. Em especial, a Matheus de Galiza, pela convivência e pelas broncas. A Kelson, pelas broncas. E a Letícia Buriti, minha irmãzinha, também pelas broncas. Obrigado por me presentear com um novo conceito de família.

Aos meus amigos internacionalistas, por todos esses anos compartilhando conhecimentos, lutas, dúvidas e paixões. Em especial, às minhas duplas Lara Pordeus, Ulisses Gomes, Cecília Fernandes, Maria Vitória e Raphael Maciel, com quem muito me orgulha ter dividido sala, trabalhos e sonhos. Agradeço também àqueles companheiros que muito me ajudaram: Caio, Deusdédite, Anna Bia, Perlyson, Cath, Laís e Paola. Acredito no potencial gigantesco de todos vocês.

Neste caminho, a minha escolha pelas RI se deu pela consideração do campo como um conhecimento que eu não queria viver sem ter e perseguir. É com alegria que nos últimos anos tomei as leituras como aprendizados e reflexões pessoais. Gostaria de agradecer à totalidade do Departamento de RI por todo o conhecimento proporcionado. Na condição de filho, irmão, sobrinho e primo de professores, confesso que a docência e a pesquisa sempre foram da minha maior estima e admiração pela transformação proporcionada.

Ao meu orientador, Pascoal, por toda a leveza e por toda a confiança depositada em mim. Agradeço a nossa parceria em pesquisa, extensão e monitoria, e pelas conversas

descontraídas entre e durante as reuniões. Obrigado por despertar sempre, e às vezes mais do que deveria, o meu impulso pelos diversos campos do conhecimento. Espero ter sido um bom orientando, e agradeço pelo respeito e pela compreensão ao longo da jornada. Aproveitando o parágrafo, agradeço também à professora Mariana Baccarini, por ter despertado em Teoria Política Contemporânea o meu interesse pelo institucionalismo.

À professora Eliane, por ter me acompanhado ao longo da graduação com muitos ensinamentos, fundamentais na minha orientação profissional e acadêmica. Sou eternamente grato desde a primeira quinta-feira de aula de Introdução à Ciência Política. Agradeço todo o apoio e a confiança depositada em cada projeto, bem como em cada crítica e cada reunião.

Aos professores Thiago e Túlio, por me honrarem compondo a banca de defesa deste trabalho, e pelos ensinamentos ao longo das disciplinas.

Na condição de institucionalista, agradeço também a Universidade Federal da Paraíba, minha segunda casa desde a infância, pela educação pública, gratuita e de qualidade. Obrigado por cada conversa, cada aula, cada chuva, cada alegria e cada um dos muitos cafés. Foi-me uma verdadeira raiz, e agradeço a todos os ombros de gigantes e a todas as asas proporcionadas pelos professores, alunos e servidores – em especial Marquinhos e Leandro -.

Por fim, agradeço a mim mesmo. Sinto que nos últimos quatro anos, me perdi no internacional para me encontrar nos diversos locais e pessoas, muitas delas listadas acima. Sou grato pelo meu eu do passado, pelos seus erros, acertos, escolhas e transformações. Sou grato pela sensação de saudade que a graduação deixará em mim, e pelo futuro que há de vir.

Obrigado a todos por cada pena que forma as minhas asas. Obrigado a cada raiz que forma o meu porto seguro.

RESUMO

O presente artigo provém de um Projeto de Iniciação Científica da UFPB, com o objetivo de criar rotinas de programação para coletar, tratar, organizar e analisar o corpus textual da pesquisa. O projeto se origina na compreensão da importância da mudança do discurso político como indicador da saliência doméstica de normas internacionais, neste caso, a agenda de desenvolvimento global expressa pelos Objetivos do Desenvolvimento Sustentável (ODS). De início, a pesquisa partiu no esforço de construir uma base de dados extensiva, contendo a totalidade dos discursos dos Presidentes da República do Brasil desde 1985, marco da redemocratização do País. Como aparato conceitual, compreende-se saliência como a variação da legitimidade das normas internacionais em um contexto doméstico, medido por meio de mudanças institucionais, políticas e de discurso, ou retóricas. Tratando sobre um período tão longo, observou-se a mudança na forma de armazenamento do acervo da Presidência da República, das quais decorreu a necessidade de criação de *scripts* específicos para automatizar a coleta e o tratamento dos dados de forma operacional para análises futuras. Como resultado, este trabalho descreve as tecnologias desenvolvidas, na forma de *scripts* na linguagem Python, para a construção de um corpus textual extenso, de mais de seis mil discursos, e adaptável para a coleta futura. Além de ultrapassar a falta de padronização e sistematização decorrentes de mudanças de governo e tecnologias, o resultado evidencia uma aproximação frutífera entre as Relações Internacionais e demais ciências sociais com elementos da computação. Inicialmente pensado para o projeto específico, o mérito desta pesquisa pode ser utilizado para diversas outras análises, de forma operacionalizável, e facilmente adequada a softwares modernos. Operacionalizada a base de dados, a pré-leitura do material motivou mudanças nas pretensões de análise, razão pela qual *scripts* para análise não foram continuados.

Palavras-chave: Coleta de dados; Tratamento de dados; Pesquisa qualitativa; Análise de Conteúdo; Discurso Político.

ABSTRACT

This article comes from a Scientific Initiation research Project at UFPB, with the objective of creating programming scripts to collect, clean, organize and analyze the textual corpus of the research. The project stems from understanding the importance of changing political discourse as an indicator of the domestic salience of international norms, in this case, the global development agenda expressed by the Sustainable Development Goals (SDGs). Initially, the research started with the effort to build an extensive database, containing all the speeches of the Presidents of the Republic of Brazil since 1985, a milestone of the country's redemocratization. As a conceptual apparatus, salience is understood as the variation in the legitimacy of international norms in a domestic context, measured through institutional, political and discourse or rhetoric changes. Dealing with such a long period, there was a change in the way of storing the collection of the Presidency of the Republic, which resulted in the need to create specific scripts to automate the collection and processing of data in an operational way for future analysis. As a result, this work describes the technologies developed, in the form of scripts in the Python language, for the construction of an extensive textual corpus, with more than six thousand speeches, and adaptable for future collection. In addition to overcoming the lack of standardization and systematization resulting from changes in government and technologies, the result shows a fruitful approximation between International Relations and other social sciences with elements of computing. Initially thought for the specific project, the merit of this research can be used for several other analyses, in an operational way, and easily adapted to modern software. Once the database was operationalized, the pre-reading of the material led to changes in the analysis intentions, which is why analysis scripts were not continued.

Keywords: Data collection; Data cleaning; Qualitative research; Content analysis; Political discourse.

SUMÁRIO

1. INTRODUÇÃO:

O presente artigo aborda a construção de uma base de dados textuais extensa, a partir da criação de rotinas de programação na linguagem Python. Inicialmente, o trabalho tem origem em um plano de Iniciação Científica (IC) cujo enfoque reside em uma visão teórica institucionalista e com tema da saliência doméstica de normas internacionais, neste caso, a agenda global de desenvolvimento materializada pela Agenda 2030. O Plano de Iniciação Científica propunha como objetivos a criação de rotinas de programação, em linguagens como R e Python – para a coleta e sistematização de uma base de dados contendo a totalidade dos discursos presidenciais a partir de 1985. O projeto teve duração de setembro de 2021 até agosto de 2022, com apresentação dos resultados em outubro do mesmo ano. Apesar da inicial intenção de se debruçar sobre a Agenda 2030, a sistematização do corpus textual revelou um enorme potencial para uma diversidade de campos das Relações Internacionais, como a Difusão de Políticas Públicas e a Análise de Política Externa (APE). Neste sentido, os rumos das futuras análises a partir dos dados se modificou para além do previsto inicialmente, o que será potencializado pela divulgação e publicação dos dados.

Lançada em 2015, a Agenda 2030 substituiu os antigos Objetivos de Desenvolvimento do Milênio (ODM), criticados por terem sido elaborados de forma tecnocrática, com pouca participação e por deixarem de fora uma série de dimensões que seriam incluídas ao longo dos diversos debates que construíram a agenda internacional dos Objetivos do Desenvolvimento Sustentável (ODS) (FUKUDA-PARR, 2016; PNUD, 2021). Elaborada a fim de ser implementada de forma integral e transversal - por meio do diálogo entre os diversos objetivos e metas - conta como característica a amplitude temática em seus 17 objetivos e 169 metas. A partir disso, enquanto nova agenda global de desenvolvimento, a Agenda 2030 e o desenvolvimento sustentável têm sido temas de amplo interesse acadêmico e da sociedade civil (PNUD, 2021).

Na medida em que a mudança no nível de saliência doméstica ocorre a níveis institucionais, de governo ou de discurso, este último foi o escolhido inicialmente para a mensuração da saliência da Agenda 2030 no Brasil. A fim de viabilizar a pesquisa, optou-se pela análise de conteúdo dos discursos, encontrados nos arquivos públicos da Presidência da República, na forma de arquivos de dados textuais não estruturados. De 1985 até os mais recentes, a biblioteca do Ministério das Relações Exteriores (MRE) disponibiliza os discursos

em distintos tipos de arquivo, não devidamente tratados para a utilização mais eficiente, como em softwares mais avançados para a análise qualitativa. Assim, ao mesmo tempo em que o corpus possibilita um amplo leque de análises, a organização e exploração do material representou um desafio à parte para a consecução da pesquisa.

Neste sentido, o presente trabalho tem como objetivo apresentar as tecnologias desenvolvidas para a solução de problemas na consecução de pesquisas na área de Relações Internacionais, aplicável às demais áreas das ciências sociais e humanas. As tecnologias apresentadas constituem rotinas de programação na linguagem Python, mais especificamente para a coleta, tratamento e sistematização do corpus textual. Considera-se que o artigo possui um enorme potencial de auxiliar outros pesquisadores das humanidades que busquem trabalhar com grandes volumes de material textual em suas pesquisas. Por suposto, as soluções descritas abaixo foram particularmente customizadas para o problema específico. Ainda assim, o conjunto de tecnologias podem ser adaptadas a diversas demandas semelhantes, de *web scraping* de dados textuais, organização e manipulação de arquivos de forma automatizada na linguagem Python (GRUS, 2019).

Em termos de resultados alcançados, se demonstrou que o uso de linguagem de programação em Python é um aliado poderoso na fase de coleta e sistematização de dados - neste caso, qualitativos - e na pré-análise, deixando um material extremamente volumoso devidamente tratado para a parte analítica. O presente artigo está organizado em três seções, para além desta introdução e das conclusões. Primeiramente, discutiu-se a importância dos discursos e a visão teórica que orientou o projeto inicial, seguido da descrição dos problemas a serem enfrentados para a coleta e tratamento dos dados textuais. Em seguida, há uma descrição das tecnologias desenvolvidas, explicando as escolhas realizadas ao longo do processo, de forma a adequar e trazer maior eficiência aos resultados. Por fim, são apresentados os resultados finais, bem como uma avaliação do emprego da programação na pesquisa.

2. OS PROBLEMAS A SEREM RESOLVIDOS:

Como apresentado na introdução deste artigo, a fim de viabilizar a pesquisa, optou-se pela análise de conteúdo dos discursos para a mensuração da saliência doméstica da Agenda 2030. O desenvolvimento da pesquisa realizada perpassa pela compreensão e importância dos discursos para analisar a difusão de políticas, ideias e normas internacionais, bem como a sua saliência no cenário doméstico. Além disso, os desafios encontrados para a operacionalização

dos dados textuais, os discursos, é um aspecto fundamental para a descrição das rotinas de programação detalhadas na seção seguinte, de acordo com os objetivos da pesquisa.

Segundo Schmidt, em seu texto seminal do institucionalismo discursivo, o discurso se refere à articulação e representação de ideias (SCHMIDT, 2008). Mais do que isso, de acordo com Hall (1993), o discurso é a representação do conhecimento pela linguagem, implicando em significado e influenciando a realidade (HALL, 1993). Carta e Narmínio (2021) argumentam que o significado do mundo material tem origem no discurso, em processos cognitivos e percepções, em seu artigo que explora a materialização de ameaças de uso da força.

O papel do discurso não pode ser ignorado para a compreensão de processos políticos e institucionais (SCHMIDT, 2008), ainda mais em uma realidade internacional cada vez mais interdependente. Aqui, é aplicável o conceito de interdependência como realidade na qual os agentes internacionais atuam e interagem na compreensão da complexidade de suas atitudes e de seus reflexos em diferentes regimes internacionais (GILARDI, 2012). Reconhecendo a importância teórica posta pela interdependência, Gilardi (2012) aproxima a literatura das Relações Internacionais da abordagem da Difusão, questionando como o contexto internacional, assim como ideias, normas e políticas promovidas internacionalmente por Estados ou por Organizações Intergovernamentais (OIs), influenciam decisões, políticas e ideias em outros países. Segundo Faria (2018), o atual contexto de interdependência implica na amplificação as potencialidades da difusão, uma vez que problemas são considerados como comuns em diferentes unidades políticas, potencializando a internacionalização de soluções e agendas (GILARDI, 2012; FARIA, 2018).

O questionamento de Gilardi (2012) e o apontamento de Faria (2018) encontram eco na visão institucionalista de Cortell e Davis (2000), em sua contribuição para operacionalizar as pesquisas de difusão internacional de normas e políticas e seus efeitos domésticos. Os autores apontam para a intensidade de determinada norma internacional em um contexto doméstico, a saliência doméstica, como fundamental para a compreensão de transformações políticas e institucionais (CORTELL, DAVIS, 2000). Neste sentido, normas salientes são aquelas identificadas como capazes de gerar sentimentos de obrigação dentro das estruturas políticas domésticas, legitimando comportamentos e políticas específicas. Ao mesmo tempo, deslegitimam escolhas alternativas, bem como desvirtuamentos em relação à norma são geralmente esclarecidos, justificados e condenados.

No entanto, a saliência de uma norma não deve ser igualada ou medida unicamente segundo o comportamento e a *compliance* do Estado. Quanto à sua mensuração, os autores argumentam um modelo de análise baseado na mudança em três níveis: no discurso político nacional, nas instituições e nas políticas públicas estatais (CORTELL, DAVIS, 2000). Dentro disso, a mudança discursiva é colocada como primeiro indicativo de oscilação da saliência das normas internacionais e, portanto, ponto chave e indicativo do rumo das demais mudanças institucionais e políticas no âmbito doméstico (CORTELL; DAVIS, 2000).

Retomando a importância do discurso, metodologias como análise do discurso ou análises de conteúdo que se debruçam sobre discursos têm sido incorporadas à Análise de Política Externa (CARTA; NARMÍNIO, 2021; HUSAR, 2016), incluindo casos brasileiros (LUSTIG, 2016; VILELA; NEIVA, 2011). Porém, o mesmo não pode ser dito do campo de difusão de políticas, no qual a necessidade de incorporação de ideias e de novos métodos de análise têm sido defendidos (BÉLAND; ORENSTEIN, 2013; GILARDI, 2016; GILARDI; WASSERFALLEN, 2017).

Diante do exposto acima, acredita-se que a automação da coleta e tratamento dos discursos presidenciais, possibilitada para abranger um período temporal significativo, que se estende de 1985 a 2022 - como disponibilizado pela Biblioteca da Presidência da República -, constitui um ponto chave na compreensão de mudanças a nível institucional e de política pública materializada.

Em primeiro lugar, a partir dos discursos é possível identificar a presença, e o nível da presença das normas internacionais, principalmente na forma de demandas por alterações nas agendas políticas e nas normas domésticas, evidenciando os níveis da saliência de normas internacionais (CORTELL, DAVIS, 2000) como a Agenda 2030 e dos ODS. Assim, o discurso político representa um fator chave para o avanço da compreensão da literatura de Difusão e mudança institucional. Além disso, a análise que recai sobre os discursos pode evidenciar mudanças na retórica e percepção sobre distintos problemas tanto entre indivíduos como em um mesmo indivíduo, reforçando também elementos de aprendizado resultados da interdependência sobre os tomadores de decisão (FARIA, 2018; DOLOWITZ, 2021). Para além disso, tal mensuração, por sua vez, abre espaço para entender os constrangimentos e limites que a agenda internacional para o desenvolvimento impõe aos atores internos e aos outros níveis de mudança.

Compreendida a importância dos discursos para as futuras análises, é possível se debruçar sobre os problemas enfrentados no desenvolvimento da pesquisa. Apesar de disponibilizados publicamente pela Biblioteca da Presidência da República, o volume de discursos é grande: mais de seis mil discursos desde o Governo Sarney, de 1985 em diante. Além disso, é preciso considerar a forma de organização dos discursos na base de dados da Presidência da República. Do início do período republicano ao final do governo Lula (2010), os discursos são disponibilizados em PDF. Do início do governo Dilma em diante, estão disponibilizados diretamente na página da Web. Tais questões são resultados nas mudanças tecnológicas e de governo, com repercussões também para o tratamento dos arquivos, como será tratado abaixo.

Ou seja, realizar o procedimento manualmente representaria três desafios: o primeiro é o volume de trabalho ao pesquisador, seja um aluno ou professor universitário; o segundo é proceder com esse download a partir de duas fontes distintas: pdf e Html[1]. Para o material disponível em página da Web, o procedimento para coleta envolve selecionar o texto, copiá-lo para área de trabalho, criar um arquivo, colar o conteúdo, salvar o arquivo conforme critério de nome - neste caso, a data do discurso formato americano[2] - e fechar. Procedimento muito mais longo e demorado do que o "clicar" em um link que dispara o download (como no caso do PDF). Tais limitações do acervo resultam em reflexos negativos no trabalho demandado para pesquisas que os utilizassem de forma sistemática, na medida em que o trabalho manual também abre maior margem para o erro (VILELA; NEIVA, 2011). Em seguida, um terceiro desafio se impõe: organizar os arquivos no computador, ou seja, criar um sistema de pastas e nomes de arquivos para que os mesmos possam ser facilmente identificados durante o processo de análise de conteúdo.

Após proceder com a coleta, um segundo conjunto de desafios se apresenta: tornar o material "analisável". Isso significa a limpeza dos discursos coletados, bem como preocupações técnicas quanto ao funcionamento dos arquivos. Quanto a limpeza, problemas foram notados relativos à separação silábica de palavras, não corrigidas apenas na coleta - deixando a palavra cortada por um hífen -, bem como a existência, nos arquivos em PDF, de capas, cabeçalhos e notas de rodapé que comprometem a integridade dos dados. Já quanto aos problemas de ordem

[1] As páginas da Web estão escritas na linguagem HTML.
[2] Mês/Dia/Ano

técnica, ressalta-se que o grande volume de documentos representou um problema também durante a exploração do material em softwares de análise qualitativa.

Mesmo que se consiga fazer o download manual dos documentos com os discursos para o computador, o pesquisador que objetiva realizar uma análise de conteúdo se depara com a necessidade de codificar e categorizar os documentos seguindo a orientação teórica proposta (BARDIN, 2009; SAMPAIO; LYCARIÃO, 2021). Optou-se por realizar a análise de conteúdo com o software Atlas.ti, e tempo e recursos financeiros foram investidos no início da pesquisa para aprender como utilizar o software conforme as demandas da pesquisa. Todavia, testes preliminares revelaram o mesmo "travava" quando grandes volumes de documentos eram utilizados, em decorrência do tamanho de arquivos PDF, prejudicando a codificação e capacidade de análise. Ou seja, foi necessário transformar os arquivos de .pdf para .txt, como forma de reduzir seu peso e manter o software utilizad. Para tanto, e pelo exposto anteriormente, o desafio do volume de documentos se impunha: como transformar quase milhares de arquivos de .pdf para .txt?

Em resumo, a fim de operacionalizar a pesquisa, seriam necessárias centenas de horas de trabalho humano altamente repetitivo, monótono e sujeito a erros inerentes à repetição. Não foi antes do início de 2020, no início da pandemia, que se vislumbrou que essas tarefas eram ideais para serem desempenhadas por "bots" programáveis. Uma aproximação com o Centro de Estudos das Negociações Internacionais (CAENI) da Universidade de São Paulo permitiu uma primeira visualização do que seria possível fazer utilizando a linguagem Python. A partir daí, e através de projetos específicos de iniciação científica como o discutido neste artigo, foram realizados estudos e, finalmente construídas as soluções descritas abaixo.

3. PROCEDIMENTOS METODOLÓGICOS: CONSTRUINDO SOLUÇÕES

Em busca da compreensão da alteração do discurso político como meio de aprender sobre a saliência da agenda internacional de desenvolvimento no cenário doméstico, os objetivos de criação de rotinas - *scripts* - em linguagem de programação se vinculam a compreensão exploração do corpus textual para a análise de conteúdo. Em seu livro "Análise de Conteúdo", Laurence Bardin (2009) a conceitualiza como:

> [...] um conjunto de técnicas de análise das comunicações que utiliza procedimentos sistemáticos e (sic) objectivos de descrição do conteúdo das mensagens indicadores (quantitativos ou não) que permitam a inferência de conhecimentos relativos às

> condições de produção/recepção (variáveis inferidas) destas mensagens (BARDIN, 2009, p.44).

De início, a referência aos conjuntos de técnica e a inferência - esta enquanto finalidade da análise - ressaltam o caráter maleável da análise de conteúdo. Para Bardin (2009), isso dificulta definir a análise de conteúdo enquanto molde metodológico único para análises com objetos, finalidades e pesquisadores distintos. Na cronologia da análise de conteúdo, o pesquisador perpassa por etapas que misturam o rigor objetivo com a subjetividade. Ainda na primeira das etapas, a pré-análise, além de escolher os documentos, cabe ao pesquisador realizar a leitura flutuante, isto é, um primeiro contato com o material para ser invadido por impressões que mais tarde auxiliarão na elaboração de hipóteses e vínculos teóricos (BARDIN, 2009).

A escolha do corpus - do conjunto de documentos que será submetido a análise - deve respeitar as seguintes regras: da exaustividade, incorporando todos os elementos do corpus, salvo por razão justificável; da representatividade, com a possibilidade de realizar a análise em uma amostra - o que não foi realizado nesta pesquisa, devido a coleta do total de discursos do período -; da homogeneidade, sendo os documentos representantes de critérios claros de escolha; e, a pertinência e adequação a análise pretendida (BARDIN, 2009). Neste sentido, observa-se a adequação da escolha dos discursos presidenciais dispostos nos arquivos públicos da Presidência da República e do Congresso Nacional. Na medida em que a compreensão da saliência internacional perpassa em compreender a alteração dos discursos ao longo do tempo dos tomadores de decisão (CORTELL, DAVIS, 2000) e observar seu apreender e seu aprendizado quanto à agenda internacional, observa a pertinência do corpus.

Além disso, a homogeneidade é preservada diante do critério de escolha pela análise de discursos dos tomadores de decisão. Quanto à exaustividade e a representatividade, é fundamental esclarecer a busca nesta pesquisa por sistematizar e coletar os arquivos públicos dos discursos presidenciais pós-redemocratização em sua totalidade. Destarte, observa-se um alto número de documentos, que apontaria a pesquisa rumo a definição de amostras, ou redução do escopo, não fossem as possibilidades apresentadas pela automação - melhor detalhada na seção de Resultados e Discussões. Desta forma, a regra de exaustividade foi preservada na busca por coletar e sistematizar a totalidade dos discursos.

Bardin (2009) coloca o uso do computador como ferramenta capaz de acelerar e de dar rigor a coleta, sistematização e a análise como um todo. Em um mundo em que cada vez mais

informações são digitalizadas e um volume cada vez maior de dados é produzido, as possibilidades acadêmicas são ampliadas (KING, 2011). É necessário progressivamente que o cientista social também seja um cientista de dados, capaz de utilizar ferramentas que ampliem seu escopo de atuação (GRUS, 2019). Neste sentido, o presente trabalho apresenta tecnologias desenvolvidas para a solução de problemas na consecução de pesquisas na área de Relações Internacionais, aplicável às demais áreas das ciências sociais, utilizando linguagem de programação em Python. No entanto, ressalta-se sempre que o computador não substitui, e não pode substituir, o elemento humano na totalidade da pesquisa (BARDIN, 2009), questão que será retomada nas seções posteriores.

De forma a desempenhar a coleta e sistematização do corpus textual da pesquisa, buscou-se na literatura sobre raspagem de dados e ciência de dados de forma geral, em Fóruns da Web e em cursos - online e de extensão -, de modo a aprender instrumentalmente as linguagens de programação. Este trabalho foi realizado a partir da orientação do coordenador do projeto.

Detalha-se a escolha pela linguagem Python pela gratuidade, portabilidade - os *scripts* e informações são facilmente compartilhados - e reconhecidas pela sua consistência (LUTZ, ACHER, 2007; GRUS, 2019). Além disso, a linguagem conta com muitas "bibliotecas" úteis relacionadas a modelagem, coleta e análise de dados (GRUS, 2019). Estas bibliotecas ampliam as possibilidades e facilitam o executar de tarefas como o *web scraping* - "raspagem da web", isto é, a coleta de dados e extração de informações de forma automatizada da internet (MITCHELL, 2019). Neste caso, a pesquisa automatizou a coleta de dados dos discursos presidenciais disponibilizados ao público nas bibliotecas virtuais da Presidência da República. Para isso, se utilizam "bibliotecas" de Python como *selenium* e *Pathlib*.

Coletado e organizado o corpus de forma mais eficiente e confiável pela automação, a pesquisa poderia prosseguir para a análise de conteúdo, em sua pré-análise, codificação do material, categorização e elaboração de inferências sobre o analisado (BARDIN, 2009).

A pesquisa alcançou com sucesso a sistematização e automação da coleta do corpus por meio da criação de rotinas de Python. Neste sentido, milhares de discursos de todos os Presidentes desde José Sarney puderam ser armazenados tanto em formato de Tabela - em arquivos xlsx e csv - como em arquivos de texto - ".txt"s -, permitindo uma organização para futuras análises. Não apenas isso, como a reutilização dessas rotinas e sua posterior adequação

permitem atualizar a base de dados quando conveniente. Na seção "Anexo" estarão colocadas as rotinas de programação criadas ao longo da pesquisa e detalhadas abaixo.

3.1 Web scraping e Organização dos discursos em Tabelas:

De início, procedeu-se automatizando a coleta dos discursos disponíveis em Html e alocando-os em uma tabela .csv e .xlsx na forma de um *data frame*[3]. Para tal, a rotina se valeu das bibliotecas *Selenium* e *Pandas*. *Selenium* é uma ferramenta de automação que permite a navegação e interação do programa escrito com o *webdriver*, ou o navegador. É por meio do *Selenium* que se automatiza a abertura do navegador, a chegada no site desejado e a manipulação da informação presente naquele site (MUTHUKADAN, 2018). É fundamental ao longo das rotinas sinalizar o que se está buscando, o que é feito pela interação possibilitada pelo *Selenium* entre o Python e a linguagem Html. Ao início do código, como procedural no Python, se evocam estas e outras bibliotecas, para que fossem utilizadas. Em sequência a evocação, a rotina prossegue para a abertura do navegador - Mozilla Firefox - e acesso ao site da biblioteca virtual da Presidência da República. Em alguns momentos na rotina se pode observar pedidos para que o código espere para prosseguir. Isso é feito pela biblioteca *time*, e é utilizada de forma a evitar erros. Ou seja, utilizou-se de pausas temporais no código para garantia de que o conteúdo desejado estivesse devidamente carregado e presente antes de prosseguir, evitando erros.

Em seguida, são criadas listas vazias, nas quais posteriormente serão colocados os títulos (*title*), datas (*dates*) e links (*urls*) de cada discurso. Tais listas funcionarão posteriormente para a criação de colunas na Planilha final. Criadas as listas, prossegue-se para a para a criação, ou definição, de uma função, denominada "*collect()*". No Python, pode-se utilizar as ferramentas próprias da linguagem para definir uma função elaborada pelo autor da rotina. Neste caso, a função criada percorre toda a página Web, identifica todos os discursos presentes e deles extrai título, data e link de acesso. A definição da função permitiu responder ao problema da falta de padronização, desta vez não relacionada aos tipos de arquivo, mas aos títulos e posicionamento das datas. Ambos se encontravam juntos em um único campo de texto,

[3] Na rotina "Web scraping discursos - Presidência", tem-se a automação exemplificada dos discursos do ex-Presidente Michel Temer, mas que com a simples troca em uma linha de código - pela troca do link a ser acessado - foi utilizada para a coleta dos discursos de outros Presidentes. A leitura das rotinas no Anexo foram facilitadas pelas linhas em que se detalha cada seção do código. Isso é possível pela facilidade promovida na linguagem Python, que desconsidera do programa tudo que segue o carácter "#". Isso permite que se detalhe o processo, o que é utilizado para observar falhas ao longo do processo de escrita do script e, neste caso, esclarecimentos sobre o próprio material.

separados na forma de hifens[4]. Além disso, alguns discursos não apresentam data, e a separação entre dia, mês e ano é realizada pelo mesmo carácter que separa data de título e nome da cidade onde o discurso foi realizado.

Assim, para a extração de dados - na forma de texto - de forma padronizada, diversos critérios foram estabelecidos e inseridos na função *collect()* de forma a evitar erros na rotina de programação e padronizar ao máximo a tabela final. Pelo uso de vários critérios, o código conseguiu com sucesso separar e organizar datas e títulos com diferentes separações. Assim que são identificados, estes elementos são adicionados às listas respectivas e seguindo a ordem de identificação, de modo que nenhum erro de ordem é cometido pelo programa. A função *collect()* também coleta os links, por meio de um atributo da página Web - o "href" - e os adiciona à lista "*urls*".

Definida a função, ela pode ser evocada ao longo do código, o que é feito logo em seguida e só então realizar o que foi programado. Ademais, a função *collect()* pode ser alterada, de modo a se adequar às diferenças numéricas de discursos entre os diferentes presidentes, por meio dos valores 'y' e 'z' e de sua constante alteração. Por meio de loops, o 'y' determina a quantidade de vezes que se realiza a coleta, e é definido segundo o número de páginas diferentes com discursos. Isso ocorre porque a Biblioteca da Presidência da República apenas exibe 30 discursos por vez, tendo um total de páginas a depender de cada presidente[5]. Com conhecimentos também de HTML, se identificou que cada link dessas páginas é o mesmo, por exceção do número no final, em que a página seguinte assume um número maior em 30 - em decorrência do número de discursos - que a anterior. Assim, configurou-se 'z' inicialmente igual a zero, gerando um *loop* dentro da função *collect* no qual ao fim da coleta dos discursos, aumenta em 30, e o navegador segue para o link da página seguinte, até a última página. Portanto, é garantido que todos os discursos de todas as páginas estejam nas listas do programa.

Coletados todos os links de discursos, o programa então prossegue criando a lista "noticias" e abrindo cada um dos discursos individualmente. Acessado a página do discurso, o

[4] i.e. Títulos como: "Brinde do Presidente da República, Michel Temer, no almoço oferecido em homenagem ao Senhor Desiré Delano Bouterse, Presidente da República do Suriname", "03-05-2018-Discurso do Presidente da República, Michel Temer, durante Cerimônia de inauguração do Hospital Notre Dame - Barretos/SP" e "18-05-2018-Discurso do Presidente da República, Michel Temer, durante cerimônia de Entrega de Unidades Habitacionais do Programa Minha Casa Minha Vida"

[5] Deste modo, se configurou 'y' igual a zero, inicialmente. No percorrer de cada página, o seu valor aumenta em 1, e é verificado se o seu valor é igual ao número de páginas (no caso do ex-presidente Michel Temer, 15 páginas. Então, quando y se iguala a 15, a função não é acionada, e o código prosseguia).

script localiza precisamente a parte textual do discurso e a adiciona a lista recém-criada. Tudo coletado, entra em ação a criação de um *Dataframe*, uma Tabela organizada pelo *Pandas*. Esta biblioteca, evocada no início do *script*, é uma forma eficiente de manipular, modelar e analisar dados. Ela possibilita organizar dados em listas na forma de tabelas, além de ler e escrever dados, juntar bases de dados, indexar, fatiar e operacionalizar dados no geral (PANDAS, 2022). Além disso, ela permite um diálogo com outros formatos de arquivo, como .csv e .xlsx, funcionalidade aqui utilizada. O *script* pode então criar uma tabela com todos os títulos, datas e textos dos discursos, bem como colar esse *dataframe* em um documento no formato Excel para melhor visualização e futura sistematização.

A organização em tabelas facilitou a identificação e visualização dos discursos. Além disso, permitiu observar a existência de erros de formatação que colocariam a precisão de análises em risco. Foi percebido que cada linha de texto se dividia como um parágrafo, separando períodos e palavras alvo de separação silábica. Isso se dava pela presença de caracteres para quebras de linha e de parágrafos ao fim de cada linha de texto. Como consequência, os discursos estavam fracionados em parágrafos não coesos, impossibilitando análises de conteúdo e a codificação efetiva do *corpus* por meio de softwares de análise qualitativa, que identificariam, ao invés de um parágrafo completo, vários parágrafos fracionados e sem coesão. Tais problemas foram resolvidos pelas rotinas.

3.2 Criação de arquivos de Texto e correção do texto:

Coletados os discursos e organizados em tabelas, seguiu-se de modo a criar arquivos .txt individuais, por ser um formato mais versátil e leve para análises posteriores. Deste modo, outra rotina, a "Txts formatados.py", teve como objetivo, a partir do resultado anterior criar um arquivo ".txt" para cada discurso, bem como nomeá-lo da forma mais organizada possível. Além disso, a rotina conserta erros de formatação que ocorrem durante a raspagem de dados e que possam dificultar a utilização dessa informação.

Quanto à questão do nome dos arquivos, a rotina criada os nomeia segundo a data do discurso. No entanto, o sistema de data escolhido não é o usual "dia/mês/ano", mas o formato de "mês/dia/ano". Essa escolha se explica pela posterior utilização da pesquisa de softwares de análise qualitativa, os quais em sua maioria adotam o padrão estadunidense, e que utilizam datas iniciadas por mês. Além do mais, uma vez que os arquivos são salvos em pastas no computador, ficam organizados em ordem crescente a partir do nome. Optou-se por criar uma pasta para cara

administração presidencial e dentro de cada uma, subpastas para cada ano da administração, a fim facilitar análises futuras e a identificação de documentos específicos.

Diferentemente da raspagem de dados, a rotina "Txts formatados.py" se utiliza de arquivos já existentes na memória do computador para criar outros arquivos - os .txt's. Para tal, foram mapeadas bibliotecas de Python que auxiliassem em identificar arquivos e manipular Planilhas. Assim, ao início do *script*, são evocadas as bibliotecas "Openpyxl", para a manipulação de conteúdo em arquivos .csv ou .xlsx, e "Pathlib", que facilita localizar a origem e o destino dos arquivos manipulados e criados ao longo da execução da rotina. Além disso, são evocados também o "*os*", usual para manipulações de arquivos pelo Python, e "*re*", que são as "*regular expressions*" que permitem criar regras de linguagem para manipulação de arquivos de forma mais fácil.

Em sequência às bibliotecas, a rotina aponta a localização da planilha que será manipulada - resultado da rotina anterior - e a pasta final onde os novos arquivos serão criados. Novamente, a fácil manipulação destes locais de arquivo permite a organização de toda a base de dados com alto ganho de produtividade. Além disso, de forma similar à rotina anterior, listas vazias foram criadas para armazenar datas, parte de datas, títulos, texto dos discursos e nome dos arquivos. Feito isso, cada uma delas é preenchida por meio de *loops*, que seleciona o material respectivo a cada lista. No entanto, dois *loops* se diferenciam, ao realizar mais do que a captura: o relativo à pasta das partes de data, que separa cada elemento[6]; e o relativo aos textos dos discursos, que são formatados, removendo quebras de linha e de parágrafo e juntando palavras alvo de separação silábica[7].

Por fim, após os *loops* de coleta, um outro *loop* junta as partes das datas no novo formato - Mês, dia, ano - e adiciona este novo formato à lista de nomes de arquivos. Em seguida, cria - na pasta de destino - um arquivo de texto vazio com a nova data como nome. Em seguida ele abre o arquivo na função de escrita, possibilitando sua edição, e nele cola título e texto de cada

[6] Neste ponto, como as datas ainda estão em DD/MM/AAAA, o código usa o caracter de '/' para dividir cada data em 3 partes, que em seguida serão reorganizadas para formar o nome do arquivo, na forma "MM.DD.AAAA". O caracter de '/' não é utilizado no nome do arquivo por limitações do sistema, que não reconhece esse caracter para nomear arquivos, provocando erro.

[7] Neste processo, preocupações podem ser formadas sobre hífens que não utilizados para separação silábica no fim de linhas e sua distinção. No entanto, o código consegue efetivamente distinguir entre hífens apenas e hífens seguidos de quebras de linha e parágrafo.

discurso, e fecha o arquivo. Criados todos os documentos, se programou que a rotina sinalizasse o fim da execução com uma mensagem sobre a criação de documentos.

3.3 Correção de problemas e automatização de .txt pelo site:

As duas rotinas acima, apesar de extremamente funcionais, apresentaram alguns problemas relacionados à utilização de tabelas, e ao limite de caracteres e informações capaz de ser armazenado em uma célula no Excel. Isso foi identificado pelo corte de discursos mais longos, o que prejudicaria a adequação da base de dados[8]. Uma solução identificada foi pular a etapa do armazenamento dos discursos nas tabelas, e criar arquivos de texto diretamente do discurso extraído do site.

Assim, foram reunidos em uma nova rotina elementos dos dois *scripts* apresentados acima, formando a rotina intitulada "webscraping HTML2TXT.py". Como pode ser identificado no anexo, a rotina apresenta incorpora ao primeiro código as pastas relacionadas à localização do arquivo - bem como a biblioteca *Pathlib* -, ao nome dos arquivos e as partes das datas - também utilizadas para os nomear. Além disso, procedendo com a definição da função *collect()* e com os *loops* para raspar os links e acessar cada um dos discursos, observa-se a ausência da formação do *dataframe* da nova rotina. Substituiu-se tal elemento por um *loop* similar ao presente na segunda rotina, mas levemente modificado. Ele cria o arquivo de texto com o nome da data em questão, nele escreve o discurso e em seguida o salva. A formatação das datas em "Mês.Dia.Ano" foi automatizada e inserida na modificação da função *collect()*. Assim, utiliza-se das listas já formadas na primeira rotina para escrever os arquivos de texto sem a formação de um arquivo ".xlsx" inicial, o que inclusive acelerou a sistematização da base de dados.

3.4 Coleta de discursos em formato PDF:

Como foi explicitado anteriormente, a organização dos discursos dos ex-Presidentes não está colocada de forma uniformizada na biblioteca da Presidência da República. A título de exemplo, enquanto os discursos do ex-Presidente Michel Temer se encontram todos disponíveis em uma mesma listagem, os discursos do ex-Presidente Luiz Inácio Lula da Silva são separados por mandato, dentro disso, por ano. Além disso, se os discursos mais recentes estão disponibilizados no corpo do site, os mais antigos, anteriores aos Governos Dilma, ainda são

[8] Cada célula de uma planilha de Excel suporta no máximo 32.767 caracteres. Fonte: Especificações e limites do Microsoft Excel - Suporte da Microsoft

disponibilizados em arquivo PDF dentro de cada link. De forma a acessar cada um destes arquivos com ganho de produtividade foi elaborada a rotina “downloadpdf.py”.

Neste caso, as mudanças de organização da Biblioteca da Presidência também alteram as necessidades do *script*. Ao baixar o arquivo, seu nome já é a data, não precisando mais coletá-la. O mesmo vale para o título, já inserido dentro do arquivo, acima do discurso. Deste modo, se redefine a função *collect()* para apenas coletar os links de cada discurso. Na execução, coletados todos os links, a rotina então segue para abrir cada um dos links e, por meio do *Selenium* já detalhado, localizar e clicar no link de *download* do arquivo, que é baixado automaticamente. No entanto, como as datas estão no formato “Dia.Mês.Ano”, torna-se necessário aproveitar *scripts* anteriores apenas para trocar as posições de dia e mês no nome dos arquivos, o que será realizado na rotina seguinte, que criará arquivos .txt, mas a partir destes arquivos PDF baixados.

3.5 Automação da conversão de discursos em PDF para .txt:

Coletados os discursos presidenciais mais antigos em PDF, o próximo passo de modo a unificar a base de dados seria a conversão destes discursos em formato .txt, como foi realizado nos discursos mais atuais, disponibilizados no corpo do Html da Biblioteca da Presidência da República. Assim, na rotina “pdf para txt automação.py”, também presente no Anexo, utilizou-se da biblioteca “pdfplumber”, facilitadora da manipulação e extração de informações de arquivos PDF, e se inicia - em sequência a evocação das bibliotecas - por direcionar o programa para a pasta com os arquivos PDF instalados.

O *script* realiza um *loop*, com a abertura sequencial do arquivo, caso seja PDF. A rotina identifica o número de páginas do documento, e é criada a variável “final”, que contém um vazio, que mais tarde será preenchido pelo texto já limpo e formatado do PDF. Novamente, dentro deste *loop*, se cria um outro *loop*, este agora que extrai o conteúdo de cada uma das páginas, removendo também o número das páginas do corpo do texto e substituindo quebras de linha e de parágrafo e separações silábicas por vazios. No entanto, foi necessário atenção à operações com PDF, que apresentam problemas não compartilhados por manipulações com Html ou com Tabelas, como a existência de texto como as páginas do PDF, ou cabeçalhos e rodapés característicos de documentos de governo, bem como capas. Devido a mudanças de governo, os pesquisadores notaram também diferenças na organização e padronização dos PDFs, o que exigiu modificações adicionais para cada caso. Exemplificando, até o fim do

Governo Fernando Henrique Cardoso, os documentos apresentavam capas, o que mudou no Governo Luiz Inácio Lula da Silva, que adicionou informações no cabeçalho e no rodapé, que também se modificaram ao longo dos anos. Questões como essas eram esperadas, devido a mudança tecnológica e de gestão ao longo dos 37 anos dos quais se buscou automatizar a coleta.

Por fim, o script cria o nome do arquivo .txt e em seguida o próprio arquivo, que será então nomeado e localizado na memória do computador. Nele, será escrito o texto do discurso já limpo. Em seguida, o arquivo é fechado, com a continuação do *loop* até operar em todos os PDFs. Deste modo, apesar das mudanças de organização, padronização e sistematização dos discursos, se atingiu o sucesso em padronizar a base de dados para posterior análise.

4. RESULTADOS DA AUTOMAÇÃO DA COLETA:

Diante das possibilidades apresentadas acima pela automação da coleta e organização dos discursos, a pesquisa obteve sucesso em sistematizar a base de dados dos discursos dos Presidentes da República desde José Sarney. A título de exemplo, apenas o ex-Presidente Michel Temer apresenta 408 discursos, enquanto o ex-presidente Luiz Inácio Lula da Silva, apenas em seu primeiro mandato, soma 1012 discursos. É perceptível, portanto, o ganho de produtividade resultante do desenvolvido acima, capaz então de destravar possibilidades de análise do material, aproximadamente 6 mil discursos, em decorrência também da limpeza dos documentos - com a identificação de separação de palavras e sua correção. Uma vez programadas e revisadas, as rotinas permitem coletar e organizar de forma automatizada o corpus textual no período de uma tarde, deixando o material extremamente volumoso pronto para a fase de codificação e análise.

Desta forma, a linguagem de programação Python se mostrou extremamente útil e versátil para as finalidades da pesquisa, sendo o aprendizado da ferramenta e de suas possibilidades aproveitados para o desenvolvimento de atividades de acordo com o plano original. Ao longo do último ano, a pesquisa conseguiu superar inúmeros desafios impostos pelo corpus escolhido, de modo a estar preparada para seguir para a etapa de análise. Apesar de prescrever inicialmente a criação de rotinas em Python e R para análise do corpus, o seguimento da pesquisa revelou possibilidades de ferramenta e de metodologia mais próprias à finalidade de compreender a saliência doméstica da Agenda 2030 no Brasil, e a outros temas diversos. Isso se relaciona justamente aos limites da atuação dos computadores que Bardin (2009) trata

em seu método, alertando para a importância de não renunciar o elemento humano (BARDIN, 2009).

Neste sentido, sistematizado o material, a realização da pré-análise, e da leitura do material sistematizado revelaram, como era previsto, uma enorme diversidade temática. Em decorrência disso, se preteriu por continuar a pesquisa dentro de uma abordagem qualitativa, saindo de uma análise baseada apenas na frequência de palavras e na correlação temporal com outros universos. Assim, optou-se por proceder na análise de conteúdo de forma mais aprofundada, ainda com o auxílio de outra ferramenta, o software de análise qualitativa "Atlas.ti". Tal software possibilita a codificação e categorização de forma ágil e automatizada, mas com facilidade para a visualização do conteúdo analisado, mostrando-se muito útil para as análises pretendidas.

5. CONCLUSÃO:

O presente artigo tecnológico apresentou a forma pela qual os pesquisadores foram capazes de sistematizar com sucesso uma base de dados que reúne todos os discursos presidenciais disponíveis em arquivo público do período de 1985 até o presente. O objetivo de criação de rotinas de programação utilizando a linguagem Python se mostrou extremamente produtivo, levando, neste caso, a uma aproximação entre os métodos de pesquisa no campo das Relações Internacionais com as Ciências da Computação. Deste modo, observa-se como cumpridas as regras de exaustividade, homogeneidade, da adequação e da pertinência do corpus às análises visadas. Para tal, foram superados diversos obstáculos relacionados à falta de padronização na organização e formatação dos discursos, sendo capaz de desenvolver o conhecimento capaz de gerar o ganho de produtividade pretendido. A busca na literatura sobre ciência de dados, em Fóruns da Web e em cursos foi fundamental para essa instrumentalização.

Como resultado, reunido o *corpus*, a pré-análise do material e a leitura flutuante foi facilitada e melhor organizada, com enorme ganho de qualidade. Os novos arquivos abrem e são processados facilmente, tornando-os viáveis a manipulação por softwares destinados à análise de conteúdo e análise de dados qualitativos, como o Atlas.ti. Isso tornou a codificação para o projeto atual e para pesquisas posteriores não apenas viável, como também mais eficiente. A partir da pré-análise do material, percebeu-se que a futura combinação e emprego de análise de conteúdo e do discurso seria beneficiada por tais softwares, inclusive na compreensão da saliência doméstica de normas internacionais.

Ficou demonstrado o enorme ganho de produtividade decorrente do aprendizado da linguagem Python e sua aplicação em pesquisa qualitativa na grande área das Ciências Sociais e, especificamente, no campo das Relações Internacionais. Fica demonstrado também que a utilização de ferramentas de programação abre novos horizontes na pesquisa social, dado o aumento exponencial na capacidade do pesquisador coletar, armazenar, tratar e analisar dados das mais diversas fontes e formas.

Observa-se um cumprimento de um objetivo mais amplo de sedimentar no curso de Relações Internacionais o estudo e a compreensão de ferramentas contemporâneas de organização e análise de dados, principalmente pelo uso da linguagem Python (GRUS, 2019). Assim, os resultados obtidos conseguiram fortalecer a pesquisa relacionada a saliência de normas internacionais no ambiente doméstico, na medida em que estão abertas as possibilidades de análise quanto à alterações nas agendas políticas, com efeitos institucionais, nas políticas públicas (CORTELL, DAVIS, 2000) e no aprendizado dos tomadores de decisão (FARIA, 2018).

Inicialmente pensado como Iniciação Científica para o desenvolvimento de pesquisas relacionadas ao desenvolvimento sustentável e a Agenda 2030, ideias desenvolvidas no decorrer do processo revelaram um potencial maior para o corpus. Foram abertas portas para diversas frentes de pesquisa futura. Por exemplo, a análise aprofundada das causas domésticas e internacionais para institucionalização de políticas e sua relação com a alteração do discurso, como colocado por Cortell e Davis (2000) e por Gilardi (2012). Além disso, o corpus textual pode ser amplamente utilizado em estudos de Política Externa Brasileira e de Análise de Política Externa (VILELA, NEIVA, 2011). Mais do que a utilização de discursos para o enriquecimento de análises, a base de dados possibilita utilizações sistemáticas, comparações e inferências sobre um longo período. Abre-se as portas para a sistematização facilitada de discursos abrangentes sobre temas recorrentes à PEB, como o Conselho de Segurança das Nações Unidas e a Amazônia, permitindo ampliar o conhecimento sobre o tema. Por outro lado, a sistematização possibilita realizar de forma exploratória pesquisas relacionadas a temas mais ofuscados, a exemplo do continente africano e a difusão de políticas de projetos específicos.

REFERÊNCIAS

ASCHER, David; LUTZ, Mark. **Learning Python**. Sebastopol: O'Reilly, 2007.

BARDIN. L. **Análise de conteúdo**. Lisboa: Editora Edições 70, 2009.

BÉLAND, Daniel; ORENSTEIN, Mitchell A. International organizations as policy actors: An ideational approach. **Global social policy**, v. 13, n. 2, p. 125-143, 2013.

CARTA, Caterina; NARMINIO, Élisa. The Human Factor: Accounting for Texts and Contexts in the Analysis of Foreign Policy and International Relations. **International Studies Perspectives**, v. 22, n. 3, p. 340-360, 2021.

CORTELL, Andrew P.; DAVIS JR, James W. Understanding the domestic impact of international norms: A research agenda. **International Studies Review**, v. 2, n. 1, p. 65-87, 2000.

DOLOWITZ, David P. Learning and transfer: who learns what from whom?. In: **Handbook of Policy Transfer, Diffusion and Circulation**. Edward Elgar Publishing, 2021. p. 26-42.

FARIA, Carlos Aurélio Pimenta de. **Políticas públicas e relações internacionais**. Brasília: ENAP, 2018.

FUKUDA-PARR, Sakiko. From the Millennium Development Goals to the Sustainable Development Goals: shifts in purpose, concept, and politics of global goal setting for development. **Gender & Development**, v. 24, n. 1, p. 43-52, 2016.

GILARDI, Fabrizio. Transnational diffusion: Norms, ideas, and policies. **Handbook of international relations**, v. 2, p. 453-477, 2012.

GILARDI, Fabrizio. Four ways we can improve policy diffusion research. **State Politics & Policy Quarterly**, v. 16, n. 1, p. 8-21, 2016.

GILARDI, Fabrizio; WASSERFALLEN, Fabio. Policy diffusion: mechanisms and practical implications. In: **Governance Design Network Workshop, National University of Singapore, Singapore, February**. 2017. p. 87.

GRUS, J. **Data Science do zero: Primeiras regras com o Python**. [s.l.] Alta Books, 2019.

HALL, Peter A. Policy paradigms, social learning, and the state: the case of economic policymaking in Britain. **Comparative politics**, p. 275-296, 1993.

HUSAR, Jörg. Framing Foreign Policy in India, Brazil and South Africa. **Switzerland: Springer**, 2016.

LUSTIG, Carola M. Soft or Hard Power? Discourse Patterns in Brazil's Foreign Policy Toward South America. **Latin American Politics and Society**, v. 58, n. 4, p. 103-125, 2016.

MITCHELL, Ryan. W**eb scraping with Python: Collecting more data from the modern web.** Sebastopol: O'Reilly Media Inc., 2019.

MUTHUKADAN, Baiju. *Selenium with Python*. **Selenium-Python**. Disponível em: <https://selenium-python.readthedocs.io/index.html>. Acesso em 20 jul. 2022.

PANDAS. **About Pandas**. Disponível em: <https://pandas.pydata.org/about/index.html>. Acesso em 15 out. 2021.

PNUD. **Guia de Territorialização e Integração dos Objetivos de Desenvolvimento Sustentável**. Brasília: PNUD, 2021. 64 p. – (Coletânea Territorialização dos ODS: Seu município ajudando a transformar o mundo).

SCHMIDT, Vivien A. Discursive institutionalism: The explanatory power of ideas and discourse. **ANNUAL REVIEW OF POLITICAL SCIENCE-PALO ALTO-**, v. 11, p. 303, 2008.

VILELA, Elaine; NEIVA, Pedro. Temas e regiões nas políticas externas de Lula e Fernando Henrique: comparação do discurso dos dois presidentes. **Revista Brasileira de Política Internacional**, v. 54, p. 70-96, 2011.

ANEXOS

Anexo I: Web scraping discursos - Presidência.py

```
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.common.exceptions import NoSuchElementException
from selenium.webdriver.remote.errorhandler import ErrorHandler
from selenium.webdriver.common.keys import Keys
from time import sleep
import pandas as pd
#Iniciar o navegador e chegar ao site do MRE
driver = webdriver.Firefox()
action = webdriver.ActionChains(driver)
driver.get("http://www.biblioteca.presidencia.gov.br/presidencia/ex-presidentes/michel-temer/discursos-do-presidente-da-republica?b_start:int=0")
WebDriverWait(driver, 10)
#Listas em que serão inseridos cada um dos atributos
title = []
dates = []
urls = []
# localizando cada um dos atributos
def collect():
   x = 0
        while x < 31:
              titulos = driver.find_elements_by_xpath('/html/body/div[2]/div[2]/div[1]/div/div[3]/div[1]/div/div[4]/article/div/h2')
              lnks = driver.find_elements_by_xpath("/html/body/div[2]/div[2]/div[1]/div/div[3]/div[1]/div/div[4]/article/div/h2/a")
      wait = WebDriverWait(driver, 10)
      for titulo in titulos:
         if len(titulo.text.split('-')) >= 6:
            separar = titulo.text.split('-')
            title.append(separar[3]+ separar [4] + separar [5])
            dates.append(separar[1]+separar[0]+separar[2])
         if len(titulo.text.split('-')) == 5:
            separar = titulo.text.split('-')
            title.append(separar[3]+ separar [4])
            dates.append(separar[1]+separar[0]+separar[2])
         if len(titulo.text.split('-')) == 4:
            separar = titulo.text.split('-')
```

```
            title.append(separar[3])
            dates.append(separar[1]+separar[0]+separar[2])
        elif len(titulo.text.split('-')) < 4:
            title.append(titulo.text)
            dates.append('sem data')
            print('sem data neste título')
    for lnk in lnks: # localizando o elemento href, em que estão os hyperlinks para cada noticia
        print(lnk.get_attribute('href'))
        urls.append(lnk.get_attribute('href'))
        x=x+1
y = 0
z = 0
while y < 15:
driver.get('http://www.biblioteca.presidencia.gov.br/presidencia/ex-presidentes/michel-temer/discursos-do-presidente-da-republica?b_start:int=' + str(z))
    if z <= 390:
        collect()
        z = z + 30
    else:
        break
print(title)
print(dates)
print(urls)
# Usando a lista de urls, pega o texto dentro de cada um dos links
noticias = list()
n = 0
for ur in urls:
    n = n+1
    driver.get(ur)
    texto = driver.find_element_by_xpath('//*[@id="parent-fieldname-text"]').text
    noticias.append(texto)
    print(n, texto)
data_frame = pd.DataFrame() # Criando o dataframe
data_frame["titulos"] = title
data_frame["datas"] = dates
data_frame['texto'] = noticias
print(data_frame)

data_frame.to_excel(r"C:\Users\Romberg de Sá Gondim\Desktop\PIVIC - Pascoal Teófilo - Mudança Institucional no Nordeste\teste\Presidencia.xlsx", index = False) #aqui ficaria o local do arquivo criado em xlsx no computador usado

#é possível colocar também a adaptaçao pra mudar de página depois de 30 noticias extraídas, sendo 30 o intervalo do site
```

Anexo II: Txts formatados.py

```
import string
import openpyxl # necessário instalar openpyxl no prompt de comando (pip install openpyxl)
from pathlib import Path #o comando path troca cada / por \ na leitura do script, e assim localiza o arquivo sem falhas
import re
import os
excel_path = Path("C:/Users/Romberg de Sá Gondim/Desktop/PIVIC - Pascoal Teófilo - Mudança Institucional no Nordeste/Documentos_Testes/Discursos.xlsx") #local onde se insere o path do excel - lembrar de inverter as barras para pegar no Pathlib
datesdmy = list()
titulo = list()
texto = list()
dates_parts = list()
file_names = list()
pasta_path = Path("C:/Users/Romberg de Sá Gondim/Desktop/PIVIC - Pascoal Teófilo - Mudança Institucional no Nordeste/Documentos_Testes/test") # pasta final dos arquivos, colar aqui seu path, lembrando de trocar as barras. !!!! importante lembrar da barra no final, para juntar com o nome do arquivo !!!!
wb_obj = openpyxl.load_workbook(excel_path)
sheet_obj = wb_obj.active
m_row = sheet_obj.max_row
for i in range(2, m_row + 1): #loop que salva as datas para depois fazer o split
    cell_obj_date = sheet_obj.cell(row = i, column = 2) #lembrar de posicionar a coluna correta para as datas (diferent do python, o openpyxl conta no excel a partir de 1, e não de 0)
    print(cell_obj_date.value)
    datesdmy.append(cell_obj_date.value)
for date in datesdmy:
    dates_parts.append(date.split('/'))
print(dates_parts)
for i in range(2, m_row + 1): #loop que salva os títulos
    cell_obj_title = sheet_obj.cell(row = i, column = 1)
    print(cell_obj_title.value)
    titulo.append(cell_obj_title.value)
for i in range(2, m_row + 1): #loop que salva os textos
    cell_obj_text = sheet_obj.cell(row = i, column = 3)
    print(cell_obj_text.value)
    texto.append(cell_obj_text.value.replace("\n", "").replace("\r", ""))
x = 0
for part in dates_parts:
    datemdy = part[1] + '.' + part[0] + '.' + part[2] #nova formatação da data para nomear o arquivo com MM.DD.AA
    file_names.append(str(datemdy))
    file_to_open = pasta_path / (datemdy + ".txt") #construção do novo arquivo, já com nome da data de cada discurso
```

```
    f = open(file_to_open, mode = 'w')
    f.write(titulo[int(x)] + '\n' + '\n' + texto[int(x)]) #inserir aqui localização das células de cada texto
    x=x+1
    f.close()
print('Documentos txt criados! :)')
```

Anexo III: webscraping HTML2TXT.py

```
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.common.exceptions import NoSuchElementException
from selenium.webdriver.remote.errorhandler import ErrorHandler
from selenium.webdriver.common.keys import Keys
from time import sleep
import pandas as pd
from pathlib import Path
#Iniciar o navegador e chegar ao site do MRE
driver = webdriver.Firefox()
action = webdriver.ActionChains(driver)
driver.get("http://www.biblioteca.presidencia.gov.br/presidencia/ex-presidentes/michel-temer/discursos-do-presidente-da-republica/discursos")
WebDriverWait(driver, 10)
#Listas em que serão inseridos cada um dos atributos
title = []
dates = []
urls = []
file_names = []
pasta_path = Path("C:/Users/Romberg de Sá Gondim/Desktop/PIVIC - Pascoal Teófilo - Mudança Institucional no Nordeste/Documentos_Testes/test") # pasta final dos arquivos, colar aqui seu path, lembrando de trocar as barras. !!!! importante lembrar da barra no final, para juntar com o nome do arquivo !!!!
# localizando cada um dos atributos
def collect():
    x = 0
    while x < 31:
        titulos = driver.find_elements_by_xpath('/html/body/div[2]/div[2]/div[1]/div/div[3]/div[1]/div/div[4]/article/div/h2')
        lnks = driver.find_elements_by_xpath("/html/body/div[2]/div[2]/div[1]/div/div[3]/div[1]/div/div[4]/article/div/h2/a")
        wait = WebDriverWait(driver, 10)
        for titulo in titulos:
            if len(titulo.text.split('-')) >= 6:
                separar = titulo.text.split('-')
                title.append(separar[3]+ separar [4] + separar [5])
                dates.append(separar[1]+'.' + separar[0]+ '.' + separar[2])
            if len(titulo.text.split('-')) == 5:
                separar = titulo.text.split('-')
                title.append(separar[3]+ separar [4])
```

```
            dates.append(separar[1]+ '.' + separar[0]+ '.' +separar[2])
        if len(titulo.text.split('-')) == 4:
            separar = titulo.text.split('-')
            title.append(separar[3])
            dates.append(separar[1]+ '.' +separar[0]+'.' +separar[2])
        elif len(titulo.text.split('-')) < 4:
            title.append(titulo.text)
            dates.append('sem data')
            print('sem data neste título')
    for lnk in lnks: # localizando o elemento href, em que estão os hyperlinks para cada noticia
        print(lnk.get_attribute('href'))
        urls.append(lnk.get_attribute('href'))
        x=x+1
#loop da coleta:
y = 0
z = 0
while y < 15: #definido pela quantidade de páginas de discurso
    driver.get('http://www.biblioteca.presidencia.gov.br/presidencia/ex-presidentes/luiz-inacio-lula-da-silva/discursos/1o-mandato/2003?b_start:int' + str(z))
    if z <= 390:
        collect()
        z = z + 30 #definido pelo número de discursos em cada página (no caso do Lula eram 20 por página, logo z= z + 20)
        y = y + 1 #de modo a criar um cálculo de quando parar, definido de acordo com a quantidade de páginas.
    else:
        break
print(title)
print(dates)
print(urls)
# Usando a lista de urls, raspagem do texto dentro de cada um dos links:
noticias = list()
n = 0
for ur in urls:
    n = n+1
    driver.get(ur)
    wait = WebDriverWait(driver, 2)
    texto = driver.find_element_by_xpath('//*[@id="parent-fieldname-text"]').text
    noticias.append(texto)
    print(n, texto)
#Loop para formação dos nomes dos arquivos, criação deles, escrita do texto do discurso e seu armazenamento na pasta devida.
for date in dates:
    file_names.append(str(date))
```

```
    file_to_open = pasta_path / (date + ".txt") #construção do novo arquivo, já com nome da data de cada discurso
    f = open(file_to_open, mode = 'w')
    f.write(title[int(x)] + '\n' + '\n' + texto[int(x)]) #inserir aqui localização das células de cada texto
    x=x+1
    f.close()
```

Anexo IV: downloadpdfs.py

```
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.common.exceptions import NoSuchElementException
from selenium.webdriver.remote.errorhandler import ErrorHandler
from selenium.webdriver.common.keys import Keys
from time import sleep
#Iniciar o navegador e chegar ao site do MRE
driver = webdriver.Firefox()
action = webdriver.ActionChains(driver)
driver.get("http://www.biblioteca.presidencia.gov.br/presidencia/ex-presidentes/luiz-inacio-lula-da-silva/discursos/1o-mandato/2003")
WebDriverWait(driver, 10)
urls = [] #Listas em que serão inseridos os links
def collect(): # definindo a coleta dos links:
    x = 0
    while x < 11:
        lnks = driver.find_elements(by=By.XPATH, value= '//*[@id="content-core"]/article/div/h2/a')
        wait = WebDriverWait(driver, 10)
        for lnk in lnks: # localizando o elemento href, em que estão os hyperlinks para cada noticia
            print(lnk.get_attribute('href'))
            urls.append(lnk.get_attribute('href'))
            x=x+1
#coletando os links:
y = 0
z = 0
while y < 15:
    driver.get('http://www.biblioteca.presidencia.gov.br/presidencia/ex-presidentes/luiz-inacio-lula-da-silva/discursos/1o-mandato/2003?b_start:int=' + str(z))
    if z <= 200:
        collect()
        z = z + 20
    else:
        break
print(urls)
#com os links encontrados, abre cada um deles e clica no link para download
for ur in urls:
    driver.get(ur)
    #WebDriverWait(driver, 2)
    element = driver.find_element(by=By.XPATH, value= '/html/body/div[2]/div[2]/div[1]/div/div[3]/div[1]/div/div[4]/p/a')
    element.click()
```

Anexo V: pdf para txt automação.py

```
import pdfplumber
import os
import re

path = r"D:\OneDrive - ccsa.ufpb.br\bases de dados qualitativo\Discursos - executivo\Presidentes da República\02 Itamar\1992"

files = os.listdir(path)

for file in files:
    if file.endswith ('.pdf'):
        arquivo = pdfplumber.open(os.path.join(path, file))
        n = len(arquivo.pages)
        final = ""
        for page in range(1, n):
            data1 = str(arquivo.pages[page].extract_text())
            data = re.sub('\d+', '', data1)
            final = final + data.strip().replace("-\n", "").replace("\n", " ").replace("\r", "")
        name_txt = file[:-4]
        x = open('{}/TXT/{}.txt'.format(path, name_txt), "w+", -1, "utf-8")
        x.write(final)
        x.close()
```