META 3 – GITHUB COM SCRIPTS PARA RASPAGEM DE DADOS EM SITES DE FONTES DE FINANCIAMENTO

TERMO DE EXECUÇÃO DESCENTRALIZADA
UnB/MCTI

Brasília
2022

1. IDENTIFICAÇÃO DO PROJETO

Nº do Processo Administrativo	SEI 01245.019636/2021-58 (MCTI) / 23106.111333/2020-32 (UnB)
Objetivo do TED:	Projeto de Pesquisa de Ciência de Dados aplicada ao Portfólio de Produtos Financeiros
Nº do TED:	Nº 8602383/2021
Período de duração do projeto:	17/12/2022 a 17/03/2023

1.2. OBJETIVO DO PROJETO

O Projeto de Pesquisa de Ciência de Dados aplicada ao Portfólio de Produtos Financeiros terá suas ações conduzidas por pesquisadores do Laboratório de Aprendizado de Máquinas em Finanças e Organizações (LAMFO), vinculado ao Departamento de Administração (ADM) da Faculdade de Economia, Administração, Contabilidade e Gestão de Políticas Públicas (FACE) da UnB.

O objetivo geral do projeto envolve a identificação e a implementação de formas inteligentes e mais eficientes de busca, tratamento, organização e visualização da informação para o portfólio de produtos financeiros. Os objetivos específicos do projeto são:

- Identificar e implementar buscas automatizadas nos endereços já visitados, com um enfoque de projeto ágil, com análise exploratória e visualização de dados;
- Identificar e implementar um processo para categorização, análise e interpretação de dados automatizadas, com um enfoque de projeto ágil;
- Identificar e implementar mecanismos de disponibilização da solução com recomendação e interação com o usuário para obter feedback e aperfeiçoar continuamente a usabilidade do instrumento, por meio de aprendizado por reforço.

2. META 3- ATIVIDADE: RASPAGEM DE DADOS EM SITES COM INFORMAÇÕES SOBRE FONTES DE FINANCIAMENTO

2.1 OBJETIVO

Disponibilizar o GitHub com os scripts e as rotinas de raspagem de dados das informações de financiamento, bem como o banco de dados de Oportunidades, Notícias, Política e Projetos. Além disso, os produtos das rotinas de raspagem apresentam o link para o acesso do site e as políticas para candidatura da instituição que for considerada validada pelo MCTI.

2.1 PREMISSAS CONSIDERADAS

A raspagem de dados foi realizada nos sites validados pelo MCTI, que tinham permissão de acesso para raspagem de dados e apresentados na língua inglesa. Dentre os sites que foram apresentados, os que têm potencial apresentam algum projeto que abarque pesquisadores ou instituições brasileiras.

2.2 DADOS DE ENTRADA

Em sua maioria, os sites utilizados como *input*, isto é, que passaram pelo processo de *scrapping*, possuem algumas características em comum: (*i)* área que descreve a política/critérios para aplicações; (*ii)* área de notícias sobre os projetos; (*iii)* área de projetos e ou oportunidades de pesquisa. No total, são 116 sites que apresentavam potencial para o portfólio financeiro do MCTI, e que foram utilizados como *inputs* no processo.

Neste sentido, os sites diferem em sua estrutura, tornando o processo de raspagem único para cada instituição. Em alguns sites temos as informações distribuídas em forma de tabela e ou com os dados sendo distribuídos em páginas diferentes e de forma ordenada ou não.

Assim, os sites utilizados possuem em sua composição, a propensão de apresentarem oportunidades de pesquisa com bolsa para os indivíduos e ou informações das instituições que pretendem se aplicar aos projetos. O apoio financeiro pode tomar diferentes formas: *grants; fellowship; scholarship*; e outras que podem variar de instituição para instituição. O objetivo é adequar os códigos de raspagem para cada site e realizar a classificação sobre os tipos de bolsas que são apresentadas em cada site. Uma outra característica relevante é que os sites escolhidos são, em sua maioria, em língua inglesa.

Os sites, portanto, permitem coletar quatro tipo de informação: (1) Oportunidades, onde temos os prospectos dos projetos; (2) Notícias, onde novas oportunidades podem ser detectadas; (3) Política, onde a instituição descreve um pouco da história, os princípios, e os valores que guiam as ações do financiador; (4) Projetos, onde o financiador divulga o tipo de atividade que está em andamento e pode sugerir o que a instituição fornecedora procura para concessão de fundos.

2.3 DICIONÁRIO TÉCNICO

Neste subitem, apresenta-se os principais tópicos do dicionário técnico: *"O que são Bibliotecas ou Módulo?"*; *"Bibliotecas Básicas"; e "Bibliotecas de WebScrapping".*

O que são Bibliotecas ou Módulo?

Uma biblioteca é uma coleção de módulos de script acessíveis a um programa *Python*, isto é, um pacote de códigos que está pronto no Python. Dessa forma, você pode instalar uma biblioteca que foi produzida por outra pessoa e utilizar as ferramentas dessa biblioteca para resolver os problemas que você está enfrentando.

Bibliotecas Básicas

Numpy

Definição: É uma biblioteca da linguagem Python, chamada de *Numerical Python*, é uma coleção de funções e operações que ajudam a executar cálculos numéricos com facilidade. O *NumPy* oferece uma biblioteca para cálculos fáceis e rápidos.

Para deixar a ideia mais clara de como funciona a biblioteca, podemos utilizar um exemplo mais prático. Suponha que você queira resolver uma equação do segundo grau. Para isto você precisa, por exemplo, ter um conhecimento das operações de soma e subtração. Tendo isso em mente, você irá atrás de um livro de matemática básica para obter tal conhecimento e conseguir executar os cálculos. Nesse exemplo, se levarmos para linguagem de programação, podemos dizer que a biblioteca *Numpy* seria o livro de matemática básica, onde se encontra o conhecimento de soma e subtração. Logo, ao utilizarmos uma função dessa biblioteca, estamos dizendo para a máquina ir nesta biblioteca e pegar um certo "conhecimento" para executar algum cálculo.

Pandas

Definição: É uma biblioteca da linguagem Python, utilizada para manipulação e análise de dados. A biblioteca permite ler, manipular, agregar e plotar os dados de forma simples.

Para exemplificarmos esta biblioteca, podemos usar o exemplo da criação de uma matriz. Digamos que você queira criar uma matriz. Para isso, você precisa manipular certos dados para encaixá-los corretamente na matriz. O *Pandas* funcionaria como o livro que contém o conhecimento necessário para manipulação. Logo, ao executar uma função da biblioteca, estamos pedindo para a máquina acessar este "livro" e executar esses conhecimentos para construir a matriz corretamente.

WebScrapping

Definição: É o ato de coletar dados estruturados na Web de maneira automatizada. Dessa forma, podemos chamar o *WebScrapping* de Raspagem de Dados ou Extração de Dados da Web - ambas definições refletem bem o que é feito pelo *WebScrapping*. Neste sentido, a Raspagem de Dados desempenha um papel fundamental ao ceder os dados que serão utilizados pelas bibliotecas *Numpy* ou *Pandas* e para outros fins.

A prática de *Webscrapping*, é uma maneira de automatizar o processo da coleta de dados, em uma certa página para uma análise posterior. Para exemplificar este caso, suponha que queremos saber quantas ofertas do produto *Macbook air* existem na página principal do Mercado Livre. Podemos fazer isto manualmente, acessando a página e contando um a um. Porém, se quisermos economizar mais tempo e obter uma resposta com uma menor margem de erro, podemos utilizar algumas sequências de códigos que basicamente diz para a máquina acessar o site e realizar esta contagem. Fazendo isso, utilizamos a capacidade de processamento da máquina para economizar tempo.

Bibliotecas de WebScrapping

urlib

Definição: É uma biblioteca para acessar, ler e fazer o *parse* (que é basicamente transformar um dado de um formato para outro) de uma URL. De certa forma, é uma biblioteca que realiza o *request* de uma URL, que possibilita a extração de dados feitas pelo BeautifulSoup.

requests

Definição: É uma biblioteca que requisita o acesso a uma URL. De certa forma, a biblioteca *requests* é considerada como *easy-to-use* ao ser comparada com a *urlib*. Apesar disso, a *urllib* é uma biblioteca que apresenta algumas funções a mais que a *requests* não apresenta.

BeautifulSoup (bs4)

Definição: É uma biblioteca para extrair dados de HTML e arquivos XML. Neste sentido, o *bs4* é uma biblioteca que necessita de bibliotecas como a *urlib* e a *requests* para poder funcionar. No geral, tem ótimos resultados e funciona de forma eficiente.

2.4 METOLOGIA

A partir da utilização da linguagem Python e das bibliotecas mencionadas na seção anterior (2.3) foi possível realizar a raspagem de dados. Os passos que foram realizados são os seguintes: Definição do diretório do projeto no código *ppfcentral*; Importação dos módulos auxiliares, nos quais as bibliotecas estão listadas na seção 2.3. A maioria é embutida no python básico, porém é necessário instalar também as bibliotecas: *pandas*; *numpy*, *bs4*; *requests*; *currencyconverter*; *googletrans*; e *lxml* que são utilizadas tanto no ppf quanto nos imports dos scrapers; Além disso, foi feito a importação automática dos módulos de Scrapping e a remoção de Scrappers com problema; Logo após, foi definido um diretório para salvar os arquivos e a criação de uma pasta para os produtos dos códigos e a criação de pastas diárias; Palavras chaves ou *Keywords* foram definidas de forma que o cenário para iniciar o código de raspagem estava pronto; O código que roda as funções é iniciado e logo após há uma função que atualiza a base de dados; Após uma verificação que é feita pela função se há novas informações na pasta *output*, os arquivos são atualizados na Base Principal; Enfim, são criadas bases aumentadas que compõe todos os sites em quatro divisões: (1) Oportunidades; (2) Notícias; (3) Políticas; e (4) Projetos.

Para informações mais detalhadas acerca desse processo, um arquivo html (relatorio_raspagem.html) com o passo a passo detalhado e comentários sobre o código será anexado juntamente a esse relatório.

2.4 MEMÓRIA DE ESFORÇO

O processo de raspagem se baseou no histórico dos arquivos e códigos pré-existentes que estavam disponibilizados no *github,* estes foram apresentados para a equipe de Raspagem de Dados.

Como primeiro passo para a realização da tarefa, a equipe estudou os códigos passados para entender qual a linha metodológica utilizada pelos pesquisadores anteriores e qual a forma de tornar os códigos, que apresentavam alguns equívocos, em códigos funcionais e objetivos.

Em seguida, foram realizadas as devidas correções nos *scraps* para tornar o código do *ppfcentral* funcional. Dessa forma, foi possível gerar uma base para cada site, aglomerando o produto de todos os dias de rotina. Por fim, foi criado um código que concatena todas as bases semelhantes, gerando quatro arquivos *.csv* que são o principal produto do esforço realizado pela equipe.

Algumas alterações pontuais e adições de novos sites devem ser realizadas ao longo do tempo, e isso se deve a natureza mutável dos sites utilizados nos dados de entrada.

2.5 RESULTADOS FINAIS

Dentre os sites apresentados pelo MCTI como válidos, foi possível adaptar e criar quatro funções, cada uma relativa a um dos escopos da raspagem. As funções são respectivamente: Oportunidade (1); Notícias (2); Política (3); e Projetos (4). Neste sentido, as bases de dados foram conglomeradas com base nas quatro funções, isto é, o script *ppfcentral* raspa os sites que possuem rotinas de raspagem, aloca os *dataframes* em pastas respectivas ao dia em que foi realizado a rotina, atualiza a Base Principal com

informações novas e, por fim, concatena todas as funções semelhantes de todos os sites. Alguns dos sites apresentados não podem ser raspados completamente, entre os motivos podemos citar: *(i)* Ausência de projetos abertos no período no qual foi arquitetado o script; *(ii)* Bloqueio para a requisição de acesso ao site; *(iii)* Estrutura do site em um estado que impossibilita a raspagem de dados de forma correta.

Com isso, o objetivo principal deste relatório foi disponibilizar o GitHub com os scripts e as rotinas de raspagem, com e sem API, de dados das informações de financiamento. A raspagem de dados foi realizada somente em sites validados pelo MCTI, com permissão de acesso para raspagem de dados e apresentados, em sua maioria, na língua inglesa. Essa raspagem de dados foi realizada até o dia 14/04/22. Esse esforço gerou o banco de dados congelado até a data de entrega pré-estabelecida.

É importante ressaltar que todos os códigos de raspagem em questão, estão disponíveis no seguinte link: https://github.com/mcti-sefip/mcti-sefip-ppfcd2020. Dentro deste repositório, os códigos estão disponíveis na *branch* intitulada de "**scraps-desenvolvimento"**.