**Relatório de criação do script para a coleta de dados do site *nuforc.org***

1. **Introdução**

O relatório tem por objetivo demonstrar as etapas percorridas para coletar posts de usuários do site *nuforc.org* entre 07 de setembro de 1997 e 07 de Agosto de 2017 , site que contém relatos de “visualização” de OVNIs – Objetos Voadores Não Identificados, a fim de gerar um arquivo CSV (comma separeted values[[1]](#footnote-1)), que posteriormente será utilizado para extração de dados relevantes.

1. **Problema**

O site *nuforc.org* disponibiliza as informações contidas no site na forma de tabelas, contidas na *tag* HTML[[2]](#footnote-2) “table”, os posts são separados por mês e ano de postagem. Essa forma de disponibilização de dados tornaria a extração de dados complexa e exaustiva.

1. **Resolução**

A fim de resolver o problema citado na seção 2, foi desenvolvido um script utilizando a linguagem de programação **Python**, e a bibliotecas **Requests, BeautifulSoup** e **Pandas**, que servem respectivamente para realizar requisições HTTP[[3]](#footnote-3) que retornam o código HTML que compõe a página, possibilitar a navegação dentro das tags HTML, e para lidar com dados de forma estatística e com base/arquivos de dados volumosos.

**1º Passo:**

Antes de iniciar a escrita dos códigos, o estudante verificou a composição das *urls* que disponibilizam os dados no site, através dessa se identificou um padrão nas *urls* ‘<http://www.nuforc.org/webreports/ndxe201910.html>’, que além do domínio do site ‘http://www.nuforc.org/’ e o conteúdo a ser acessado ‘webreports/’, contém uma string chamada ‘/ndxe’, o ano e o mês da publicação ‘201910’ e a extensão da página ‘.html’.

**2º Passo:**

Importar as bibliotecas a serem utilizadas para a resolução do problema.



Fig1. Importando as bibliotecas.

**3º Passo:**

Após se identificar esse padrão, foi iniciado o desenvolvimento do script.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

Fig2. Criação da função para realizar *web scraping[[4]](#footnote-4)*.

O estudante optou por utilizar uma função que recebe dois argumentos, uma *string* que equivale a *url* do site de forma modificada, sem o ano e o mês de postagem e sem a extensão (i.e. ‘http://www.nuforc.org/webreports/ndxe’), e uma tupla que recebe o ano e mês de início e o ano e mês de final (i.e. (199707, 199807)).

Na função *web\_scraper*, primeiro é feito um loop que vai percorrer a tupla com o ano e mês inicial e final, dentro do loop é realizada uma verificação para identificar se o mês não é igual a zero e se o mês é menor ou igual a 12, caso verdadeiro, é realizado uma requisição do tipo GET para a url recebida concatenada ao ano e mês das postagens e a extensão da página.

O retorno da requisição GET com o uso do requests.get é enviado para a função find\_all(‘table’).text da biblioteca BeautifulSoup que retorna os dados contidos na tag HTML *table,* e a cada iteração o resultado é enviado para a função read\_html do pandas que transforma a tabela em um dataframe, após isso o dataframe que contem a tabela de dados é concatenado a um dataframe vazio que recebe todos os dataframes gerados pelo loop.

**4º Passo:**

Por fim chamamos a função web\_scraper criada e passamos como argumentos a ano e mês inicial e final e convertemos o dataframe resultante em um arquivo CSV utilizando a função to\_csv da biblioteca **pandas**.

Uma imagem contendo captura de tela

Descrição gerada automaticamente

­­Fig3. Chamando a função web\_scraper e convertendo o retorno para o arquivo csv.

1. Valores separados por virgula. [↑](#footnote-ref-1)
2. Linguagem de Marcação de Hipertexto. [↑](#footnote-ref-2)
3. Protocolo de Transferência de Hipertexto. [↑](#footnote-ref-3)
4. Extração de dados de páginas web. [↑](#footnote-ref-4)