

Relatório sobre a atividade “8 - Prática: Web Scraping com Python p/ Ciência de Dados (II)”

Lucas Gabriel Arenhardt

1. Introdução

Nesse módulo é apresentado o conceito do web scraping, que consiste em extrair informações de websites através de softwares ou scripts. No contexto da ciência de dados, a capacidade de retirar informações de diversas fontes online é essencial, permitindo a construção de grandes conjuntos de dados.

Para o desenvolvimento dessa atividade, foi utilizada a linguagem de programação Python, juntamente com as bibliotecas BeautifulSoup e Request. No vídeo, o script realiza uma busca em um site de empregos, porém a atividade aqui desenvolvida tem a finalidade de buscar um produto no site Mercado Livre e fornecer algumas informações.

2. Desenvolvimento

Primeiramente deve-se importar as bibliotecas já citadas. Para realizar as operações envolvendo o site, é criada a função “find_product()”. Nessa é utilizada a função “requests.get()”, com o site desejado de parâmetro, para armazenar o HTML da página na variável “html_text”. Logo após, a função “BeautifulSoup” é chamada para transformar o conteúdo de “html_text” em um objeto “BeautifulSoup”, permitindo analisar o HTML. Então a função soup.find_all() é usada para encontrar todos os itens da página a ser analisada, sendo os parâmetros escolhidos de acordo com a estrutura do HTML.



Imagem 1 - Parte do código contendo a utilização das funções “requests.get()”, “BeautifulSoup()” e “soup.find_all()”.

Após essa parte inicial, um “for” percorre a lista de itens e realiza diversas operações com a função item.find(), de forma a extrair os dados de nome do produto, vendedor, preço, desconto e se o frete é grátis. Nota-se que em alguns momentos a consulta pode retornar o valor “None”, nesse caso, é utilizado uma estrutura “if-else” para manipular de forma correta esses dados. Todas essas informações

coletadas são impressas com uma “f string” ao final de cada iteração. Aqui é encerrado o conteúdo da função “find_product()”.

```

1  def find_product():
2      html_text = requests.get('https://www.mercadolivre.com.br/ofertas#nav-header').text
3      soup = BeautifulSoup(html_text, 'lxml')
4      items = soup.find_all('li', class_='promotion-item')
5
6      for item in items:
7          item_name = item.find('p', class_='promotion-item__title').text
8
9          if product_name in item_name:
10             seller_name = item.find('span', class_='promotion-item__seller')
11
12             if seller_name == None:
13                 seller_name = 'Não Informado'
14             else:
15                 seller_name = seller_name.text[3:]
16
17             reais = item.find('span', class_='andes-money-amount__fraction').text
18             cents = item.find('span', class_='andes-money-amount__cents andes-money-amount__cents--superscript-24')
19             if cents == None:
20                 cents = '00';
21             else:
22                 cents = cents.text
23
24             discount = item.find('span', class_='promotion-item__discount-text').text[:3]
25
26             delivery = item.find('span', class_='promotion-item__pill')
27             if delivery == None:
28                 delivery = ''
29             else:
30                 delivery = delivery.text
31
32             print(f'''
33             {item_name}
34             Vendido por:{seller_name}
35             Preço: R${reais},{cents}
36             Desconto: {discount}
37             {delivery}''')

```

Imagem 2 - Código da função “find_product()”

Ao final, um “while: True” mantém o programa rodando constantemente, pedindo ao usuário que digite o nome do produto a ser buscado e em seguida realizando a consulta. Caso o usuário digite “*”, o programa é encerrado.

```

1  if __name__ == '__main__':
2      while True:
3          print('Digite o nome do produto que deseja buscar (ou * pra encerrar): ')
4          product_name = input('>')
5          if product_name == '*':
6              break;
7          print('Buscando...')
8          find_product()

```

Imagem 3 - Parte final do programa.

3. Conclusão

A atividade foi essencial para a introdução aos conceitos de web scraping e a sua aplicação prática, tendo como resultado um script funcional que é capaz de realizar consultas de acordo com o interesse do usuário no site Mercado Livre.