

# Raspagem de Matérias do Omelete

Alunas: Ana, Isadora e Sara

## Objetivo

O projeto utiliza web scraping para extrair dados do site Omelete. O objetivo é coletar automaticamente os títulos e links das matérias na página inicial, organizando essas informações em formato JSON.

## Metodologia

O processo foi dividido em etapas:

1. **Análise do Site:** O código HTML do <https://www.omelete.com.br/> foi inspecionado para identificar como os títulos e links das matérias estavam estruturados.
2. **Criação do Script:** Foi desenvolvido um script em Python usando Flask (para atuar como um servidor) e BeautifulSoup (para ler o HTML).
3. **Execução:** O script acessa a URL do Omelete, "lê" a página, e extrai o texto e o link de todas as tags de notícia.
4. **Filtragem:** Os dados são tratados para garantir que apenas matérias válidas (com título e link) sejam retornadas, evitando repetições ou anúncios.

## Código do Script (Backend Python/Flask)

Este é o script que faz o trabalho de extração. Ele cria um pequeno servidor que, ao ser acessado, busca as notícias no Omelete e as retorna como JSON.

Python

```
from flask import Flask, request, jsonify
import requests
from bs4 import BeautifulSoup

app = Flask(__name__)

@app.route("/extrair", methods=["POST"])
def extrair():
    url = request.form.get("url")
    try:
        page = requests.get(url)
        soup = BeautifulSoup(page.text, "html.parser")
        materias = []

        # Loop para encontrar todos os links
        for link in soup.find_all("a"):
            titulo = link.get_text().strip()
            href = link.get("href")

            # Filtro para pegar apenas notícias válidas do Omelete
            if titulo and href and "omelete.com.br" in href:
                materias.append({"titulo": titulo, "link": href})

        return jsonify(materias)

    except Exception as e:
        return jsonify({"erro": str(e)})

if __name__ == "__main__":
    app.run(debug=True)
```