The same libraries were used when webscrapping was performed for hotels

In [ ]:

```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In this case it took a larger amount of headers to access the site information.

In [ ]:

```
headers = {
    "Access-Control-Allow-Origin": "*",
    "Access-Control-Allow-Methods": "GET",
    "Access-Control-Allow-Headers": "Content-Type",
    "accept": "*/*",
    "accept-encoding": "gzip, deflate",
    "accept-language": "en-GB,en;q=0.9,en-US;q=0.8",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4
}
url = "https://www.booking.com/searchresults.pt-pt.html?aid=375654&label=msn-jrwrFdUb9zKNuCHIkGmz2g-8074541083442
response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.content, 'lxml')
```

To access the respective comments of each hotel it was necessary to build a new link to be performed the webscrapping, for this was used a common part for all links and added the name of the hotel, to access your comments specifically

In [ ]:

```
reviews_links = []
for link in soup.findAll('a', {'class': 'fb01724e5b'}):

    a = link['href']
    hotel = a.split('/')[5].split('?')[0]
    a = 'https://www.booking.com/reviews/pt/hotel/' + hotel
    reviews_links.append(a)
```

In this last part, the comments relating to each website were extracted and the .csv of each of them was created.

In [ ]:

```
count = 0
allreviews = []

for link in reviews_links:
    try:
        response2 = requests.get(link, headers=headers)
        soup2 = BeautifulSoup(response2.content, 'lxml')
        for r in soup2.findAll('span', {'itemprop': 'reviewBody'}):
            try:
                rev = r.text
                allreviews.append(rev + '\n')
            except:
                pass
    except:
        pass
    count += 1
    if allreviews != []:
        seen = set()
        allreviews = [item for item in allreviews if not(
            tuple(item) in seen or seen.add(tuple(item)))]
        dfr = pd.DataFrame.from_dict({'Avaliações': allreviews})
        print(dfr)
        dfr.to_csv('hotel' + str(count) + '.csv')
        allreviews = []
```