

## Importando as libs

```
In [2]: # Libs para web scraping
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from bs4 import BeautifulSoup
import time

# Lib para manipulação de dados
import pandas as pd
```

## Instalando o chrome driver

```
In [3]: service = Service(ChromeDriverManager().install())
```

```
[WDM] - Downloading: 100%[██████████]
| 6.79M/6.79M [00:01<00:00, 6.15MB/s]
```

## Conectando ao site

```
In [12]: # Abre o google chrome
driver = webdriver.Chrome(service=service)

# Entra no site do Linkedin
driver.get('https://www.linkedin.com/jobs/search?trk=guest_homepage-basic_guest_nav')
time.sleep(5)
```

## Preenchendo as informações de pesquisa

```
In [13]: # Cargo de interesse
cargo = 'Analista de dados'

# XPATH do Local onde digitamos o cargo de interesse
xpath_cargo = '/html/body/div[1]/header/nav/section/section[2]/form/section[1]/input'

# Digita o cargo no Local indicado
box_cargo = driver.find_element(By.XPATH, xpath_cargo)
box_cargo.send_keys(cargo)
```

```
In [14]: # País de interesse
pais = 'Brasil'

# XPATH do Local onde apagamos o que já vem escrito por padrão
xpath_clear = '/html/body/div[1]/header/nav/section/section[2]/form/section[2]/button[1]'
driver.find_element(By.XPATH, xpath_clear).click()

# XPATH do Local onde digitamos o país de interesse
xpath_pais = '/html/body/div[1]/header/nav/section/section[2]/form/section[2]/input'

# Digita o país no local indicado
box_pais = driver.find_element(By.XPATH, xpath_pais)
box_pais.send_keys(pais)
time.sleep(2)

# XPATH que seleciona o país indicado
xpath_selecao_pais = '//*[@id="location-1"]'
```

```
driver.find_element(By.XPATH, xpath_selecao_pais).click()
time.sleep(5)
```

## Carregando a tela

```
In [15]: # Coleta o tamanho da página
height_inicil = driver.execute_script('return document.body.scrollHeight')

# Looping infinito para carregar toda a página
while True:
    # Leva para o final da página
    driver.execute_script('window.scrollTo(0, document.body.scrollHeight)')
    time.sleep(3)

    # Coleta o tamanho da página
    height_final = driver.execute_script('return document.body.scrollHeight')

    try:
        # Clica no botão de carrega mais vagas
        driver.find_element(By.XPATH, '//*[@id="main-content"]/section[2]/button')
        time.sleep(3)
        # Leva para o final da página
        driver.execute_script('window.scrollTo(0, document.body.scrollHeight)')
        # Coleta o tamanho da página
        height_final = driver.execute_script('return document.body.scrollHeight')
    except:
        pass

    # Compara se houve mudança no tamanho da página
    if height_inicil < height_final:
        height_inicil = height_final
    else:
        break
```

## Coletando os dados

```
In [16]: # Coletando o html da página
soup = BeautifulSoup(driver.page_source, 'lxml')
all_vagas = soup.find('ul', {'class': 'jobs-search__results-list'})
box_vagas = all_vagas.find_all('li')

# Cria um dataframe
df = pd.DataFrame(columns = [
    'Link Vaga',
    'Link Empresa',
    'Cargo',
    'Nome Empresa',
    'Local',
    'Tempo de Abertura',
    'Status'
])

# Looping para coletar cada informação da vaga
for vaga in box_vagas:
    try:
        link = vaga.find('a', {'class': 'base-card__full-link absolute top-0 right-0'})
    except:
        link = vaga.find('a', {'class': 'base-card relative w-full hover:no-underline'})

    try:
        cargo = vaga.find('span', {'class': 'sr-only'}).text.strip()
    except:
```

```

        cargo = vaga.find('h3', {'class': 'base-search-card__title'}).text.strip()

    try:
        linkedin_empresa = vaga.find('a', {'class': 'hidden-nested-link'}).get('href')
    except:
        linkedin_empresa = vaga.find('a', {'class': 'base-card relative w-full hover:opacity-100'}).get('href')

    try:
        nome_empresa = vaga.find('a', {'class': 'hidden-nested-link'}).text.strip()
    except:
        nome_empresa = vaga.find('h4', {'class': 'base-search-card__subtitle'}).text.strip()

    local = vaga.find('span', {'class': 'job-search-card__location'}).text.strip()

    try:
        tempo = vaga.find('time', {'class': 'job-search-card__listdate'}).text.strip()
    except:
        tempo = vaga.find('time', {'class': 'job-search-card__listdate--new'}).text.strip()

    try:
        status = vaga.find('span', {'class': 'result-benefits__text'}).text.strip()
    except:
        status = ''

    # Adiciona as informações no dataframe
    df.loc[len(df)] = [
        link,
        linkedin_empresa,
        cargo,
        nome_empresa,
        local,
        tempo,
        status
    ]

```

In [18]: `df.head()`

Out[18]:

	Link Vaga	Link Empresa	Cargo	Nº Emp
0	<a href="https://br.linkedin.com/jobs/view/analista-de-...">https://br.linkedin.com/jobs/view/analista-de-...</a>	<a href="https://br.linkedin.com/company/kangu-br?trk=p...">https://br.linkedin.com/company/kangu-br?trk=p...</a>	Analista de Dados JR	Ka
1	<a href="https://br.linkedin.com/jobs/view/analista-de-...">https://br.linkedin.com/jobs/view/analista-de-...</a>	<a href="https://br.linkedin.com/company/fraport-brasil...">https://br.linkedin.com/company/fraport-brasil...</a>	Analista de Dados JR	Fra Bra Pi Ale
2	<a href="https://br.linkedin.com/jobs/view/data-analyst...">https://br.linkedin.com/jobs/view/data-analyst...</a>	<a href="https://br.linkedin.com/company/mundiale?trk=p...">https://br.linkedin.com/company/mundiale?trk=p...</a>	DATA ANALYST	Munc
3	<a href="https://br.linkedin.com/jobs/view/analista-de-...">https://br.linkedin.com/jobs/view/analista-de-...</a>	<a href="https://br.linkedin.com/company/canalag?trk=pu...">https://br.linkedin.com/company/canalag?trk=pu...</a>	Analista de dados	Ca
4	<a href="https://br.linkedin.com/jobs/view/analista-de-...">https://br.linkedin.com/jobs/view/analista-de-...</a>	<a href="https://br.linkedin.com/company/hdi-seguros?tr...">https://br.linkedin.com/company/hdi-seguros?tr...</a>	Analista de Dados Junior	Segu

Salva o dataframe na pasta que está o arquivo python no formato xlsx

```
In [ ]: df.to_excel('scrapng_linkedin.xlsx', index = False)
```