# APÊNDICE C – Código Exemplo: Robô de Coleta Preço Histórico da Ação WEGE3

### Bot Coleta WEG Stock Prices

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.firefox.options import Options

from bs4 import BeautifulSoup

from datetime import date

from dateutil.relativedelta import relativedelta

import time

import html5lib

import pandas as pd

import openpyxl

import time

import os, re

## Anotar o horário de coleta das informações

coletatime = time.strftime('%d.%m.%Y %Hh%M')

print('Coleta inciada em '+coletatime)

# Definir onde está o driver

options = Options()

options.headless = True

driver = webdriver.Firefox(executable\_path='D:/Google Drive/1. UFSC/1. Graduação/20.2/Mono II/1. TCC (Mono2)/bots/geckodriver.exe',options = options)

# Renderizar a página e receber as informações necessárias

url = "https://br.investing.com/equities/weg-on-ej-nm-historical-data"

# Abrir o site

driver.get(url)

# Dar tempo para a página renderizar

time.sleep(3)

# Clicar em aceitar cooies

try:

    accCookiesBtn = driver.find\_elements\_by\_xpath('//\*[@id="onetrust-accept-btn-handler"]')[0]

    accCookiesBtn.click()

    print("cookies button clicked")

    time.sleep(15)

except:

    pass

# Fechar o popup de login

try:

    loginCloseBtn = driver.find\_elements\_by\_xpath('/html/body/div[6]/div[2]/i')[0]

    loginCloseBtn.click()

    print("login popup closed button clicked")

    #time.sleep(5)

except:

    pass

# Expandir o horizonte de busca

## Encontrar uma data 6 meses atrás

six\_months = date.today() + relativedelta(months=-6)

startday=str(six\_months.day)+'/'+str(six\_months.month)+'/'+str(six\_months.year)

## Realizar uma consulta inciando 6 meses atrás

datePickerBtn = driver.find\_elements\_by\_xpath('//\*[@id="widgetFieldDateRange"]')[0]

datePickerBtn.click()

time.sleep(1)

startDateField = driver.find\_elements\_by\_xpath('//\*[@id="startDate"]')[0]

startDateField.clear()

startDateField.send\_keys(startday)

applyBtn = driver.find\_elements\_by\_xpath('//\*[@id="applyBtn"]')[0]

applyBtn.click()

time.sleep(1)

# Extrair o conteudo que queremos com o beautifulsoup

soup = BeautifulSoup(driver.page\_source, 'html5lib')

datatable = soup.find('table', class\_ = "genTbl closedTbl historicalTbl")

headers = []

# Encontrar os cabeçalhos da tabela

for th in datatable.thead.find\_all('th'):

    headers.append(th.text)

# Encontrar as linhas da tabela

rows = datatable.tbody.find\_all('tr')

# Criar uma lista de dicionários para guardar as informações da tabela

table\_data = []

# Extrair os dados das linhas

for row in rows:

    t\_row = {}

    # Iterar os itens de cada lista, marcando a qual cabeçalho estes são correspondentes

    for td, th in zip(row.find\_all('td'),headers):

        t\_row[th] = td.text.replace('\n','').strip()

    table\_data.append(t\_row)

# Criar os dataframes

weg\_stock = pd.DataFrame(table\_data)

# Fecha o browser depois de ter coletado informações de todas as páginas

driver.quit

# Escrever o dataframe no Excel

planilhaPath = 'D:/Google Drive/1. UFSC/1. Graduação/20.2/Mono II/1. TCC (Mono2)/bots/Scraps/stocks.xlsx'

fileExists = os.path.isfile(planilhaPath)

sheetname = 'WEG\_'+coletatime

if fileExists:

    book = openpyxl.load\_workbook(planilhaPath)

    options = {}

    options['strings\_to\_formulas'] = False

    writer = pd.ExcelWriter(planilhaPath, engine='openpyxl', options = options)

    writer.book = book

    weg\_stock.to\_excel(writer, sheet\_name = sheetname, index=False, header=True)

    writer.save()

else:

    weg\_stock.to\_excel(planilhaPath, engine= 'openpyxl',index = False, header=True,sheet\_name= sheetname)

print('Coleta realizada com sucesso. O Resultado foi salvo em '+planilhaPath)