RPA e Python: Um processo Otimizado de Webscraping.

Daniele Paulino, Matheus Mariano, Orientador: Jose Alexandre Ducatti, Coorientador Paulo Sérgio Gaudêncio Mauro

Matheus_pm97@yahoo.com.br
paulino.87@gmail.com
jose.ducatti@fatec.sp.gov.br
paulo@fatecriopreto.edu.br

Resumo: O meio coorporativo busca cada vez mais a otimização dos processos para melhorar o tempo e a qualidade dos serviços, a fim de substituir processos manuais de raspagem de dados (*Webscraping*), por meio de RPA (*Robot Process Automation*) em linguagem *Python*. Este artigo visa a automatização de processos rotineiros e repetitivos do dia a dia utilizando o misto de algumas bibliotecas em *Python* para obter dados de uma plataforma web. O trabalho mostra os impactos gerados com a atomização desses processos, assim como a redução de custos da operação (hora homem), evitando assim erros humanos e liberação de horas de trabalho.

Palavras-chave: Algoritmo, Python, RPA (*Robot Process Automation*), Automação, Processos, *Webscraping*

Abstract: The corporative environment is looking more and more for optimizing processes to improve the time and quality of services, keeping forward replace manual processes of web scraping to RPA (Robot Process Automation) using python language. This article should deliver the automation of routine and repetitive process using the mix of some libraries at python to catch data from a web platform. The study shows the impacts made with automation of these processes, as the reduction of costs labor hour, avoiding human mistakes and time free of job.

Keywords: Algorithm, Python, RPA (Robot Process Automation), Automation, Process, Webscraping

1 Introdução

Nas empresas, os processos administrativos que necessitam de análise crescem a cada dia, com exemplos como o cruzamento de dados e extração de relatórios. O grande volume de informações faz com que o colaborador disponha de um período de tempo maior para obter os dados necessários, e a característica repetitiva destes processos contribuem para que ocorram falhas humanas e inconsistências.

Em vista da necessidade da automatização de processos repetitivos, foi criado o RPA (*Robotic Process Automation*). Este processo por meio de softwares comandados por códigos de computador, formados por algoritmos que irão executar ações programadas que imitam a mesma interação do usuário com as aplicações de forma inteligente, compensando assim a carga de trabalho do colaborador. O RPA segue uma sequência de passos idênticos aos do usuário, quando opera um programa de computador para realizar um determinado processo. Os principais benefícios do RPA são agilidade no processo executado, maior assertividade, que gera menor desperdício, e maior produtividade.

Neste estudo se tratando de redução de tempo e otimização de processos de relatórios, a ferramenta escolhida foi *Python*. Que em meio a vários modos de sua utilização, há a possibilidade de uso como um robô computacional, de forma autônoma e baseada na instrução

da programação, a fim de realizar o *Webscraping* ou raspagem de dados da web por meio de RPA de forma otimizada e ágil.

2 Justificativa

Visando otimizar o tempo e recursos humanos com certas atividades repetitivas e padronizadas, as ferramentas de otimização de processos surgem para tentar solucionar esse problema obtendo assim eficiência e controle de resultados. Assim com RPA, utilizando as bibliotecas do Python, pode ser possível melhorar essa demanda.

Esse estudo visa agregação de valor e fornecer informações para a comunidade acadêmica e corporativa a respeito do tema abordado.

3 Objetivo(s)

Este artigo tem por objetivo estudar o processo básico que podemos seguir para para montagem de um RPA para raspagem de dados de um sistema Web utilizando a linguagem Python.

Objetivo Geral:

A) Reduzir o tempo gasto com extração de dados;
B) Reduzir fator humano em processos repetitivos e sistemáticos.

Objetivos específicos:

A) Estudar o desenvolvimento de um algoritmo que colete dados.
B) Analisar o uso de Python na automatização de processos.

4 Fundamentação Teórica

Estão envolvidos neste artigo 3 áreas de interesse, RPA, Python e Webscraping.

4.1 RPA

Inventado por volta do ano 2000, a Automação Robótica de Processos (Robot Process Automation ou RPA) está transformando a maneira como as empresas funcionam. O RPA visa automatizar processos, por meio de softwares comandados por códigos de computador, formados por algoritmos que irão executar ações programadas, mas de uma forma que envolva menos interferência humana que as estratégias tradicionais de automação, ele permite que você envie ações do mouse e do teclado para as janelas de diálogos e controles do Windows, automatizando tarefas como geração de arquivos ou mesmo envio de e-mails. As tarefas repetitivas e exaustivas no ambiente de negócios, tende a ir diminuindo o seu tempo de execução e custos operacionais, aumentando a produtividade e evitando erros.

4.2 Python

O Python surgiu na década de 1990 e foi criado pelo programador holandês *Guido van Rossum*. O nome é em homenagem ao grupo de comédia britânico chamado Monty Python. O objetivo era que essa linguagem fosse fácil e intuitiva, boa o suficiente para ser competitiva no mercado; ser *open source*, ou seja, aberta a quaisquer contribuições no desenvolvimento; de

fácil compreensão, e utilizável para todos os tipos de complexidade. E de fato ela atende toda essa proposta, pois é de fácil utilização e compreensão do usuário.

4.3 Webscraping

Webscraping ou raspagem de dados da web, é o processo automatizado de coleta de dados estruturados da web, que podem ser pegos manualmente, mas a ideia é a extração de forma automatizada coletando um maior volume de dados em menos tempo.

5 Trabalhos Similares

Automação de processos de negócio utilizando robotic process automation (rpa) em um centro de serviços compartilhados (csc): um estudo de caso (golçalves, 2021). Esse estudo tem o intuito de avaliar a implementação da tecnologia Robotic Process Automation (RPA) em um Centro de Serviços Compartilhados (CSC) para automações de dois processos de negócio. Entender a metodologia de gestão e controle da área responsável pelo RPA na empresa.

Extração de dados com web scraping para análise da variação de preço de veículos automotores (borges, ganimi, 2018). Portanto, este trabalho propõe o desenvolvimento de uma ferramenta de obtenção de dados de Websites – procedimento conhecido como Web Scraping – com os quais, consultas e análises podem ser feitas a fim de apoiar o processo de decisão. Como caso de uso, uma amostra de dados foi obtida do site da FIPE1 , da qual alguns gráficos serão apresentados.

Automação de processos na logística em um centro de distribuição de cervejas utilizando programação em python 3.0 (chagas, 2022). Reduzir o tempo de demanda das atividades rotineiras do colaborador proporcionando maior produtividade para o mesmo. Para a automatização destes processos são utilizadas ferramentas computacionais. Mas quando trata-se de redução de tempo e otimização de processos de relatórios, uma destas ferramentas denomina-se Python (PYTHON, 2022), e em meio a vários modos de sua utilização, há a possibilidade de uso como um robô computacional, que de forma autônoma e baseada na instrução da programação, realiza os mesmos processos que são feitos manualmente com o mouse e o teclado do computador.

6 Metodologia

A metodologia empregada neste estudo tem caráter qualitativo que se dá diante uma pesquisa sobre o estudo do processo de melhoria em extração de dados web.

A pesquisa deu-se devido o dia-a-dia de uma empresa, termos que entrar em uma página da web e buscarmos a atualização da situação de vários “Ids”, que são o meio de busca e interação na plataforma, onde o usuário deve acessar esta plataforma, inserir o ID e copiar a informação para uma planilha em Excel, montando assim um base de dados atualizado.

Assim analisamos a possibilidade dessa base ser feita automaticamente e esses Ids serem formulados em planilha padrão, onde constem os Ids a serem consultados, e o algoritmo retorne toda a pesquisa que anteriormente seria feita manualmente.

6.1 Tipo de pesquisa

A pesquisa será qualitativa e exploratória pois os dados foram coletados e analisados diretamente no ambiente em que ocorrem.

6.2 Métodos a serem utilizados

Analisaremos a possibilidade de automatizar esse processo através de um algoritmo RPA que possa fazer este mesmo processo de forma limpa, clara e objetiva, através do Python.

6.3 Ferramentas e tecnologias utilizadas

Notebook Virtual e Bibliotecas do Python.

6.4 Recursos materiais

Os recursos possivelmente utilizados serão o computador e internet, e o prompt do Jupyter notbook, eles serão utilizados como a base principal onde o algoritmo de RPA deverá executar a automação programada. Para que isso ocorra o usuário deve dar o start no processo.

6.5 Plano de trabalho

Atividade 1: levantamento de etapas: <Analisar todas as etapas feitas no processo >

Atividade 2: coleta de dados: <Coletar uma amostra dos Ids que vamos utilizar na pesquisa>

Atividade 3: Montagem do programa: <Esta etapa será a mais demorada, nela devemos codificar todas as etapas do levantamento e fazer com que o processo seja eficaz e traga os dados necessários de forma dinâmica>

7 Desenvolvimento

A pesquisa envolve uma tarefa que se inicia com uma base em Excel contendo “IDS”, esses IDS são o meio de comunicação entre o usuário e a plataforma web, é através dele que são feitas pesquisas para averiguar a situação atual de cada caso e transportar para a planilha.
Utilizamos o prompt do Jupyter notebook para os testes.
Diante disto:

1) Iniciamos com a biblioteca Pandas para visualizar a tabela em Excel, que se encontra em uma pasta de trabalho.

Código inserido no Prompt:

```
!pip install Pandas

import pandas as pd

# a linha abaixo abre a pasta de trabalho onde possui o arquivo Excel com a lista de Ids a serem pesquisados na Web, informamos o caminho.
tabela = pd.read_excel("ArquivoSalvo.xlsx")
#comando usado para mostrar os dados
display(tabela)
```

	ID	DESCRIÇÃO	CONTRATO	NUMERO MED. DRAFT	NUMERO MEDIÇÃO	OS
0	437488	NaN	NaN	NaN	NaN	NaN
1	419209	NaN	NaN	NaN	NaN	NaN
2	419259	NaN	NaN	NaN	NaN	NaN
3	421663	NaN	NaN	NaN	NaN	NaN
4	422596	NaN	NaN	NaN	NaN	NaN
5	424340	NaN	NaN	NaN	NaN	NaN
6	424367	NaN	NaN	NaN	NaN	NaN

Figura1

Figura 1 – Resultado exibido após executar os codigos mencionados acima.

2) Após isso, o proximo passo é abrir o endereço Web para consulta.
Para esta etapa utilizamos a Biblioteca **Selenium,** que permite que o Python abra seu navegador e execute os comandos.
Instalamos tambem a Biblioteca **ChromeDriver** que possibilita a comunicação do Selenium com o Google Chrome e a Biblioteca **webdriver-manager** que permite o Selenium controlar o seu navegador.
Requests é usado para fazer a requisição no site de forma que um software possa conversar com outro e permitir a troca de informações entre eles.

Bs4 ou Beautiful Soup é um pacote Python para analisar documentos HTML e XML. Ele cria uma árvore de análise para páginas analisadas que podem ser usadas para extrair dados de HTML, o que é útil para web scraping.

Código inserido no Prompt:

Instalação de Bibliotecas:

```
!pip install Selenium
!pip install ChromeDriver
!pip install webdriver-manager
!pip install requests
!pip install bs4
```

Importação:

```
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.action_chains import ActionChains
from bs4 import BeautifulSoup
import requests
import re
```

1) Abaixo utilizamos o código para gerar um looping, onde, ao buscar no arquivo em Excel, o ID ele traga todas as informações listadas raspando os dados na Web e após isso passe para o 2º ID,e assim consecutivamente até o utlimo ID, após isso ele fechará o navegador e salvará o arquivo.
   O código a seguir vai fazer uma chamada GET para a página web que queremos e criar um objeto BeautifulSoup com o HTML da página **navegador.get(url)**

Código inserido no Prompt:

```
servico = Service(ChromeDriverManager().install())
navegador = webdriver.Chrome(service=servico)

        #copiar o primeiro ID
        for i, idosp in enumerate(tabela["ID"]):
            url = f"colocar o endereço aqui"
            navegador.get(url)
            osp = navegador.find_element(By.ID,'tudo').text
            page_content = navegador.page_source
            site = BeautifulSoup(page_content, "html.parser")
            bloco = site.find('fieldset', attrs={'ui-corner-all'})
            lista = bloco.find_all('td')
            tabela.loc[i, "OS"]  = lista[4].text
            tabela.loc[i, "CONTRATO"]  = lista[6].text
            tabela.loc[i, "DESCRIÇÃO"] = lista[16].text
            tabela.loc[i, "NUMERO MED. DRAFT"]  = requisicao[14].text
            tabela.loc[i, "NUMERO MEDIÇÃO"]  = requisicao[15].text
```

```
tabela.to_excel("ArquivoSalvo2.xlsx")
tabela = pd.read_excel("ArquivoSalvo2.xlsx")
display(tabela)
navegador.close()
```

O Resultado obtido é salvo em uma nova planilha e pode ser visualizado na plataforma do código conforme podemos observar na Figura 3, os dados principais foram protegidos.

ID	DESCRIÇÃO	CONTRATO	NUMERO MED. DRAFT	NUMERO MEDIÇÃO	OS
437488	[illegible]	[illegible]	close	close	OS NÃO GERADA\t\t\t\t
419209	[illegible]	[illegible]	\n ...	Número 478334 -	130236207\t\t\t\t
419259	[illegible]	[illegible]	\n ...	\n ...	130236208\t\t\t\t
421663	Aband. Alivio - Cenario 4[illegible]	[illegible]	\n ...	Número 476832 -	130236230\t\t\t\t
422596	Aband. Alivio - Cenario 4[illegible]	[illegible]	Número 479580 - Única	RELATÓRIO DE ACEITAÇÃO\t\t\t\t	130236231\t\t\t\t
424340	[illegible]	[illegible]	Número 475307 -	Anterior	130236251\t\t\t\t
424367	[illegible]	[illegible]	Número 478479 -	Número 479692 - Parcial - 1ª Medição	130236232\t\t\t\t

Figura 3

Figura 3 – Resultado visual do Excel salvo após o código ser rodado.

8 Resultados e Discussões

Como podemos observar no desenvolvimento, com poucas bibliotecas em Python conseguimos extrair dados da Web de forma bem rápida, os dados ainda podem ser tratados para melhor utilização, melhorando ainda mais o resultado.
O tempo gasto nesse processo de forma manual é de +ou - 1min por ID. Na pesquisa de exemplificação utilizamos uma base com 7 IDS o que levaria em torno de 6 a 7 Min, o mesmo processo feito de forma automatizado retornou os dados em menos de 1 Min. Isso mostra que o tempo gasto foi reduzido em 700%.

9 Conclusões

Podemos concluir que a implementação de RPA em ambiente empresarial para automatizar tarefas repetitivas de extração de dados Web é benéfica, visto que a demanda de tarefas que se encaixam nesse padrão é alta, e o resultado da pesquisa utilizando a metodologia exemplificada neste artigo e utilizando dos processos básicos com as bibliotecas em Python para otimizar o processo desejado foi positivo. O tempo gasto para a obtenção dos dados foi significativamente 80% menor que o mesmo trabalho feito de forma manual, assim obtivemos o resultado esperado, e poderíamos otimizar os demais processos com ganho de tempo evitando falhas e erros humanas e o tempo total gasto poderá ser reduzido e adequando o código conforme a necessidade do usuário.

Agradecimentos

Agradecemos a nossa família pelo apoio, nosso orientador e professor pela ajuda.

Referências

Automação de Processos de Negócio Utilizando Robotic Process Automation (RPA) em Um Centro de Serviços Compartilhados (CSC): Um Estudo de Caso. Orientador: João Batista Simão. 2021. 68 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Universidade Federal de Uberlândia, Universidade Federal de Uberlândia, 2021. Disponível em: https://repositorio.ufu.br/bitstream/123456789/33758/1/AutomacaoProcessosNegocio.pdf Acesso em: 2 nov. 2022.

Automação de processos na logística em um centro de distribuição decervejasutilizandoprogramaçãoempython3.0. Orientador: Prof. Charles Martins Diniz. 2022. 37 f. Trabalho de conclusão de curso (Curso Bacharelado em Engenharia Mecânica) - Instituto Federal de Educação Ciência e Tecnologia de Minas Gerais, Arcos-MG, 2022. Disponível em: https://www.ifmg.edu.br/arcos/cursos-1/graduacao-1/repositorio-de-tcc/Monografia_TCC_MardenFinalizado.pdf
Acesso em: 2 nov. 2022.

Extração de dados com web scraping para análise da variação de preço de veículos automotores. Orientador: Jean de Oliveira Zahn. 2018. 53 f. Trabalho de Conclusão de Curso (Curso de Tecnologia em Sistemas de Computação) - UNIVERSIDADE FEDERAL FLUMINENSE, NITERÓI, 2018. Disponível em: https://app.uff.br/riuff/bitstream/handle/1/8930/TCC_THIAGO_DA_CUNHA_BORGES__E__ZEUS_OLENCHUK_GANIMI.pdf?sequence=1
Acesso em: 2 nov. 2022.