



Ingeniería en Computación

Curso:

Introducción Al Desarrollo De Paginas Web

Investigación III:

Web scraping

Integrantes:

Carlos Enrique Fonseca Villalobos

Luis Armando Castillo López

Campus San Carlos

Table of contents

| | |
|------------------------------------|-----------|
| Introduction | 3 |
| Web Scraping | 4 |
| Legal implications of Web Scraping | 4 |
| Existing Tools for Web Scraping | 4 |
| Languages for web scraping | 4 |
| Applied Web Scraping | 5 |
| Results | 6 |
| Code implementation in Python: | 6 |
| Database objects: | 8 |
| Conclusions | 9 |
| Bibliography | 10 |

Introduction

The objective to be accomplished in this project is a continuation of the two past researches regarding web scraping, through the application of one of the techniques, language and tools investigated, in this case Python. Making use of the knowledge acquired from the XPATH in class and what was learned in the first part of this research, it was possible to execute a program that allows a web scraping of the information of the used vehicles for sale in the VEINSA MOTORS CR website. In addition, all the data obtained on used vehicles will be stored in a database.

Web Scraping

Web Scraping, like other data technology practices, is a technique for collecting datasets on websites by extracting data or content from HTML code and storing it in databases for later use.

It is often used to obtain a large amount of information in an automated way. This information can be very useful for scraping data on a website, analyzing competitor's websites, improving marketing or SEO resources, for marketing and decision making. This practice can also be used for malicious purposes such as copyright theft or manipulation of extracted data.

- Some of the most notable uses are:
- Extracting data through the API's used.
- Extracting data and storing it in another location
- Manipulating the extracted content in order to modify it
- Analyze web structures

Legal implications of Web Scraping

Web scraping is legal if it does not violate privacy. Data mining behind login walls is similar to public data mining. But yes, this is unethical if done without permission, as it violates privacy laws.

This privacy issue affects not only the individuals who are customers of a platform, but also the owners of the platform because, according to Krotov and Silva, "Just as individuals have a right to privacy, organizations also have a right to maintain the confidentiality of certain aspects of their operations." (2)

Depending on the information "scraped" from some company, this act may imply the violation of a company's privacy rights since as users or first timers in this area we are not always aware of a company's rights.

Existing Tools for Web Scraping

While some of the existing tools can be web applications, as mentioned in the section on Web scraping techniques, there are others that would be used at the programming level, in this case selenium will be used.

Languages for web scraping

One of the most popular languages for Web Scraping is Python but even so it is not exclusive for the use of this practice, almost any language can be used for Web Scraping as long as it can lift the execution of a web browser, among them are:

Applied Web Scraping

The project was carried out through the creation of a Python function that manually takes the XPATH of the data searched in the used car sales page of the company VEINSA. The data obtained from the cars through this code would be the model, year of the car, mileage, transmission type, fuel type, traction type, displacement, passenger capacity, regular price and 6 extras of the vehicle.

The process by which the page is accessed is with a virtual environment within the project in which the libraries of "requests lxml autopep8", "Beautifulsoup" and the tool "Selenium" were installed to perform the processes by means of a ChromeDriver.

By means of the web driver used by selenium, the program goes to the page in question, enters each car that has not been sold or reserved and checks the XPATH sent for each piece of data to be obtained. This data is initially stored in a list which allows the data to be retrieved directly. The XPATH of each of the data sought would be as follows:

1. Modelo: `/html/body/form/div/div/div[1]/div[1]/div[1]/h1`
2. Año: `/html/body/form/div/div/div[1]/div[1]/div[2]/div[1]`
3. Kilometraje que tiene al ingresar al sistema:
 - a. `/html/body/form/div/div/div[1]/div[1]/div[2]/div[2]`
4. Transmisión del vehículo: `/html/body/form/div/div/div[1]/div[1]/div[2]/div[3]`
5. Combustible: `/html/body/form/div/div/div[1]/div[1]/div[2]/div[4]`
6. Tracción: `/html/body/form/div/div/div[1]/div[1]/div[2]/div[5]`
7. Cilindrado: `/html/body/form/div/div/div[1]/div[1]/div[2]/div[6]`
8. Capacidad: `/html/body/form/div/div/div[1]/div[1]/div[2]/div[7]`
9. Precio: `/html/body/form/div/div/div[1]/div[1]/div[3]/div[1]`
10. Imagen: `/html/body/form/div/div/div[1]/div[3]/div/div/div[2]/img`

These would be the extras, which are in a dynamic list in succession and are not always the same, this amount is the minimum in all entries:

- a. `/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[1]`
- b. `/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[2]`
- c. `/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[3]`
- d. `/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[4]`
- e. `/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[5]`

Subsequently, for the storage of these data obtained from the page, Firebase was used as a database, for this was added to the project the connection between python with Firebase, where the collection of data obtained by webscraping will be stored in the database, and saved with a unique id.

Results

Code implementation in Python:

```
1  from firebase import firebase
2  from selenium import webdriver
3  from selenium.webdriver.common.by import By
4  from selenium.common.exceptions import NoSuchElementException
5  import time
6
7
8  db=firebase.FirebaseApplication('https://webscrap-python-default-rtdb.firebaseio.com/')
9
10
11  link_detalles = []
12
13  # Search game in Amazon
14  def search():
15      url = 'https://veinsausados.com/buscar/'
16
17
18      # Selenium
19      options = webdriver.ChromeOptions()
20      options.add_argument('--incognito')
21      options.add_argument('headless')
22      options.add_experimental_option('excludeSwitches', ['enable-logging'])
23
24      driver = webdriver.Chrome(
25          |   executable_path="WebScraping_Python\chromedriver.exe", options=options)
26      driver.get(url)
27
28      time.sleep(3)
29
30
31  # Cantidad de páginas
32  paginacion = driver.find_elements(By.CLASS_NAME, 'search-box__page-number')
33
34  # Navegar entre páginas
35  index = 1
36  while index <= len(paginacion):
37      link_detalles = []
38      url = url + '?pagina=' + str(index)
39      driver.get(url)
40
41      autos = driver.find_elements(By.CLASS_NAME, 'random-vehicles__vehicle')
42      for auto in autos:
43          if(auto_vendido(auto, 'random-vehicles__sold') == False):
44              link = auto.find_element(By.CLASS_NAME, 'random-vehicles__vehicle-link').get_attribute("href")
45              link_detalles.append(link)
```

```

46     for link in link_detalles:
47         driver.get(link)
48
49         modelo = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[1]/div[1]/h1')
50         anno = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[1]/div[2]/div[1]')
51         km = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[1]/div[2]/div[2]')
52         transmission = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[1]/div[2]/div[3]')
53         combustible = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[1]/div[2]/div[4]')
54         traccion = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[1]/div[2]/div[5]')
55         cilindrado = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[1]/div[2]/div[6]')
56         capacidad = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[1]/div[2]/div[7]')
57         precio = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[1]/div[3]/div[1]')
58         img = driver.find_element(By.XPATH, '/html/body/form/div/div/div[1]/div[3]/div/div/div[2]/img')
59         eqBasico1 = driver.find_element(By.XPATH, '/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[1]')
60         eqBasico2 = driver.find_element(By.XPATH, '/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[2]')
61         eqBasico3 = driver.find_element(By.XPATH, '/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[3]')
62         eqBasico4 = driver.find_element(By.XPATH, '/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[4]')
63         confort1 = driver.find_element(By.XPATH, '/html/body/form/div/div/div[2]/div[1]/div/div/div/div/li[5]')












```

```



64
65         data={
66             'modelo' : modelo.text,
67             'anno' : anno.text,
68             'km' : km.text,
69             'transmision' : transmission.text,
70             'combustible' : combustible.text,
71             'traccion' : traccion.text,
72             'cilindrado' : cilindrado.text,
73             'capacidad' : capacidad.text,
74             'precio' : precio.text,
75             'img' : img.text,
76             'eqBasico1' : eqBasico1.text,
77             'eqBasico2' : eqBasico2.text,
78             'eqBasico3' : eqBasico3.text,
79             'eqBasico4' : eqBasico4.text,
80             'confort1' : confort1.text
81         }
82
83         db.post("/autosUsados",data)
84
85
86
87         url = 'https://veinsausados.com/buscar/'
88         index += 1
89

```

Database objects:




WebScrap-Python ▾

Ir a la documentación  

Realtime Database












Datos Reglas Copias de seguridad Uso

 Protege tus recursos de Realtime Database contra los abusos, como fraudes de facturación o suplantación de identidad. [Configurar la Verificación de aplicaciones](#) ✕



🔗 <https://webscrap-python-default-rtdb.firebaseio.com>

```
https://webscrap-python-default-rtdb.firebaseio.com/
├── autosUsados
│   ├── -N3KyFg1TVK7esu3rjJp
│   ├── -N3KyHgH9rMvoi2E9uF5
│   ├── -N3KyIWW0eMcSpZa3-Fc
│   ├── -N3KyJ0tVPQDxA-Uwd4R
│   ├── -N3KyJZ2VYvoEmgRKH
│   ├── -N3KyM1uR49UrXUFE8QT
│   ├── -N3KyOQsxnDT0ow4t0wK
│   ├── -N3KyP15cgKKv3ttZJXg
│   ├── -N3KyQvjQ14Yq47RgKak
│   ├── -N3KyR_ZitwLwjxP7v-i
│   ├── -N3KyTEEyqPZp8E0kJeb
│   ├── -N3KyTmk8hJkHEDWH8QZ
│   └── -N3KvUE0c0UYMFKIB3GE
```

📍 Ubicación de la base de datos: Estados Unidos (us-central1)



WebScrap-Python ▾

Ir a la documentación  

Realtime Database

Datos Reglas Copias de seguridad Uso

🔗 <https://webscrap-python-default-rtdb.firebaseio.com>

```
├── autosUsados
│   ├── -N3KyFg1TVK7esu3rjJp
│   │   ├── anno: "2013"
│   │   ├── capacidad: "4 pasajeros"
│   │   ├── cilindrado: "1000 c.c."
│   │   ├── combustible: "Gasolina"
│   │   ├── confort1: "Tapicería: Tela"
│   │   ├── eqBasico1: "Cierre Central"
│   │   ├── eqBasico2: "Vidrios Electricos"
│   │   ├── eqBasico3: "Desempañador Trasero"
│   │   ├── eqBasico4: "A/C"
│   │   ├── img: ""
│   │   ├── km: "154,939 km"
│   │   ├── modelo: "Citroen C1"
│   │   ├── precio: "Precio regular $6,500"
│   │   └── traccion: "4X2"
│   └── -N3KvUE0c0UYMFKIB3GE
```

📍 Ubicación de la base de datos: Estados Unidos (us-central1)

Conclusions

Although web scraping can be very easy to perform, there are too many variables to take into account, such as updates to the website to be scrapped that could change the web addresses from which the information is obtained or custom XPATHs for each piece of data to be extracted. In addition.

Bibliography

- (1) R. Diouf, E. N. Sarr, O. Sall, B. Birregah, M. Bousso and S. N. Mbaye, "Web Scraping: State-of-the-Art and Areas of Application," *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 6040-6042, doi: 10.1109/BigData47090.2019.9005594.
- (2) Krotov, V., & Silva, L. (2018). Legality and ethics of web scraping.
https://www.researchgate.net/profile/Vlad-Krotov/publication/324907302_Legality_and_Ethics_of_Web_Scraping/links/5aea622345851588dd8287dc/Legality-and-Ethics-of-Web-Scraping.pdf
- (3) ChromeDriver - WebDriver for Chrome. <https://chromedriver.chromium.org>