# WEB SCRAPING

# NIVEL 1: SCRAPY CON STACKOVERFLOW

EDUARD LARA

# 1. INDICE

# 1. INTRODUCCION SCRAPY

- Scrapy no es sólo una librería, es un marco de trabajo.
- Es una de las librerías más relevante en el mundo de la programación para web Scraping
- Un código de Scrapy para extraer datos se verá así:

```
from scrapy.item import Field, Item
from scrapy.spiders import Spider
from scrapy.selector import Selector
from scrapy.loader import ItemLoader

class Dato(Item):
    texto = Field()

class SpiderDeDatos(Spider):
```

# 1. INTRODUCCION SCRAPY

- Primero es la importación de módulos funciones y clases.
- Después tenemos dos clases principales.
- La primera clase Dato es una representación de los datos que queremos extraer.
- Esta clase podría representar un artículo de noticia, un producto, un usuario, un comentario, etc
- Estas entidades tienen sus propias propiedades, es decir un producto tiene precio, nombre, calificación
- Los campos dentro de nuestra clase Dato representan la información que queremos extraer de mi entidad.
- La clase Dato puede tener tantos fields o campos como queramos y con cualquier nombre.

# 1. INTRODUCCION SCRAPY

- La segunda clase es la más importante y es la clase que va a orquestar la extracción también llamada spider

```
class SpiderDeDatos(Spider):

    name = "MiPrimerSpider"
    start_urls = ['https://paginaParaExtraerDatos.com']

    #REGLAS DE DIRECCIONAMIENTO
    def parse(self, response):
        sel = Selector(response)
        titulo_de_pagina = sel.xpath('//h1/text()').get()
        print (titulo_de_pagina)
        lista = sel.xpath('//div[@id="datos"]')
        for elemento in lista:
            item = ItemLoader(Dato(), elemento)
            item.add_xpath('texto', './/h3/a/text()')
            yield item.load_item()
```

# 1. INTRODUCCION SCRAPY

- Primero se definen variables de clase, en este caso:
  - name: nombre de mi spider (cualquier nombre
  - URL semilla: para hacer la extracción de datos.
- A continuación se definen una serie de reglas de direccionamiento u orquestador. Esta clase es como un director de orquesta que le dice a nuestra araña extractora de datos, a través de reglas, hacia dónde tiene que ir en búsqueda de los datos
- Todo esto se refleja en la función parse.
- Recibe como parámetro response que es donde se encuentra mi árbol HTML de la url semilla. Scrapy da mecanismos para parsear el árbol
- A los paseadores de árbol scrapy se les llama selectores

# 1. INTRODUCCION SCRAPY

- Con ellos podemos buscar con XPath por Id o clase y obtener la información del elemento a extraer los datos
- También podemos obtener listas de elementos a través del selector, iterarlas y empezar a cargar los datos a través de la primera clase creada Dato
- Lo hacemos instanciando una clase llamada cargador de objetos itemloader que recibe un objeto a cargar (la clase Dato) y el selector o el árbol HTML donde vamos a buscar la información de los ítems que voy a cargar
- En este caso el campo texto de mi item es llenado con lo que se ubica en la expresión XPath
- Finalmente la función Parse necesita un yield, que es como return, que va a hacer que el ítem que he cargado con datos se pueda guardar en un archivo.

# 2. EXTRACCION STACKOVERFLOW

- Extraeremos el titulo y la descripción de cada una de las preguntas de la pagina principal de Stackoverflow.
- Anteriormente lo hicimos con Request para hacer el requerimiento y Beautiful soup para parsear el árbol HTML. Esta vez usaremos la librería Scrapy tanto para hacer el requerimiento como para pasear la respuesta.
- Primero tenemos que definir mi clase de abstracción de los ítems que queremos extraer. Nuestro ítem es cada una de las preguntas y sus propiedades son:
  - El título de la pregunta
  - La descripción de la pregunta
- Ya tenemos definido nuestro ítem y ya tenemos definidas sus propiedades.

# 2. EXTRACCION STACKOVERFLOW

**Paso 1**. Definimos la primera clase que es la abstracción de los datos que queremos extraer. Determina los datos que tenemos que llenar y que estarán en el archivo generado

```
from scrapy.item import Field, Item
from scrapy.spiders import Spider
from scrapy.selector import Selector
from scrapy.loader import ItemLoader

class Pregunta(Item):
    pregunta = Field()
    descripcion = Field()
```

Llamaremos a la clase Pregunta. Esta clase debe de heredar de Item (ponemos entre paréntesis la clase de la cual queremos que herede nuestra clase)

El cuerpo de la clase es simplemente la definición de cada una de las propiedades que queremos extraer:

- Pregunta
- Descripción

# 2. EXTRACCION STACKOVERFLOW

**Paso 2.** Definiremos la clase core de scrapy que es la cual va a hacer los requerimientos y el parseo

```
class StackOverflowSpider(Spider):
    name = "MiPrimerSpider"
```

La llamaremos StackOverflowSpider. Es importante que herede de la clase Spider, en este caso porque vamos a hacer extracción de una sola página. En otro tipo de extracciones no heredamos de Spider

Primero definiremos un nombre a nuestro Spider

# 2. EXTRACCION STACKOVERFLOW

**Paso 3.** Ahora definimos el encabezado user-agent en Scrapy. Es importante definirlo siempre.

```
custom_settings = {
    'USER_AGENT': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,'+
    'like Gecko) Ubuntu Chromium/71.0.3578.80 Chrome/71.0.3578.80 Safari/537.36'
}

start_urls = ['https://stackoverflow.com/questions']
```

- En Scrapy, se define dentro del objeto custom_settings y como clave se va a usar USER_AGENT en mayúsculas
- También definimos la URL semilla donde vamos a hacer la extracción de datos, dentro de la lista start_urls, la cual solamente va a tener un elemento porque vamos a hacer extracción de una sola página

# 2. EXTRACCION STACKOVERFLOW

**Paso 4**. Definiremos la función parse donde se va a realizar el parseo del árbol HTML. Se va a llamar después de que Scrapy haga el requerimiento a la URL semilla

```
def parse(self, response):
```

- self es el 1º parámetro de la función que representa la instancia de la clase en python
- response contiene el árbol HTML.

Scrapy sólo necesita la URL semilla para hacer el requerimiento automáticamente y obtener el árbol HTML. No usa ninguna expresión explícita (como requests.get).

Aquí realizamos el parseo (2º paso) puesto que ya se ha realizado automáticamente el requerimiento (1º paso)

# 2. EXTRACCION STACKOVERFLOW

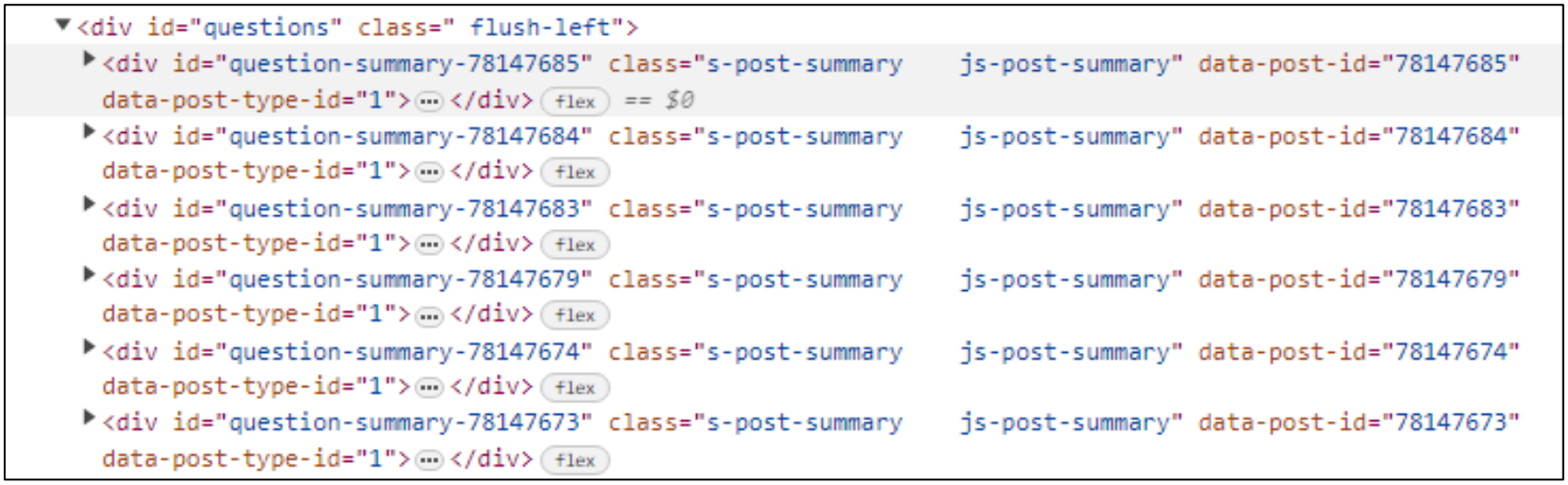
**Paso 5.** Ya no vamos a utilizar Beautiful Soup ni LXML. En Scrapy utilizaremos la clase Selector para parsear los árboles HTML.

```
def parse(self, response):

    sel = Selector(response)
```

Creamos un Selector que recibe como parámetro response, y lo guardomos en una variable. Este selector es el que nos va a servir para poder hacer consultas a la página mediante XPath (también se puede utilizar selectores CSS, otro tipo de expresiones para poder ir recorriendo los diferentes elementos del árbol)

# 2. EXTRACCION STACKOVERFLOW

**Paso 6.** Inspeccionamos la pagina de peguntas de Stackoverflow. Tenemos que obtener todos los divs de la clase s-post-summary que están dentro del div contenedor con Id questions




Tenemos que montar una expresión que obtenga cada uno de estos divs dentro de una lista para poder luego iterarlos y obtener la información que quiero de cada uno.

# 2. EXTRACCION STACKOVERFLOW

**Paso 7.** El selector permite definir una expresión Xpath,

```
preguntas = sel.xpath('//div[@id="questions"]//div[contains(@class,"s-post-summary ")]')
```

Le indicamos que en todo el árbol nos busque el Div con id questions y luego a partir de ese div, nos obtenga todos los hijos div que tengan como clase s-post-summary

El resultado de esta expresión va a ser una variable lista llamada preguntas

# 2. EXTRACCION STACKOVERFLOW

**Paso 8.** Podemos recorrer la lista, iterando pregunta a pregunta. Para cargar la información dentro de Scrapy, usaremos la clase ItemLoader, la cual va a cargar nuestros ítems. Recibe como 1º parámetro una instancia de la clase que tiene nuestra abstracción de lo que queremos extraer, la clase Pregunta, y como 2º parámetro el selector HTML donde va a estar la información con la que voy a llenar los campos de la clase Pregunta

```
for pregunta in preguntas:

    item = ItemLoader(Pregunta(), pregunta)
```

Lo guardaremos en una variable llamada ítem

# 2. EXTRACCION STACKOVERFLOW

**Paso 9.** Tenemos que llenar las propiedades de nuestro ítem Pregunta. a través de expresiones Xpath que buscan dentro del selector "pregunta". Primero llenaremos el campo pregunta, mediante la función add.xpath

```
for pregunta in preguntas:
    item = ItemLoader(Pregunta(), pregunta)

    item.add_xpath('pregunta', './/h3/a/text()')
```

Recibe dos parámetros:

- El primero es la propiedad pregunta que queremos llenar.
- El segundo parámetro es una expresión xpath que partiendo del selector con el que hemos instanciado ItemLouder, cargará la propiedad pregunta. Ponemos el XPath que dentro del div s-post-summary nos lleva al título de la pregunta → .//h3/a/text()

# 2. EXTRACCION STACKOVERFLOW

**Paso 10.** Ahora con la descripción hacemos lo mismo:

```
for pregunta in preguntas:
    item = ItemLoader(Pregunta(), pregunta)
    item.add_xpath('pregunta', './/h3/a/text()')

    item.add_xpath('descripcion', './/div[@class="s-post-summary--content-excerpt"]/text()')
```

- Primero ponemos el campo descripción
- Segundo, la expresión XPath que me lleva hasta la información de ese campo, partiendo del selector o elemento HTML con el cual instanciamos ItemLoader.

  . → Indicamos que es relativo al elemento selector

  // → Busca en todo el elemento HTML, cualquier div cuya clase sea s-post-summary--content-excerpt y de esto obtiene el texto.

# 2. EXTRACCION STACKOVERFLOW

**Paso 11.** Una vez que tenemos cargados todas las propiedades (pregunta y descripción) de nuestro ítem, tenemos que hacer una especie de return llamado yield

```
for pregunta in preguntas:
    item = ItemLoader(Pregunta(), pregunta)
    item.add_xpath('pregunta', './/h3/a/text()')
    item.add_xpath('descripcion', './/div[@class="s-post-summary--content-excerpt"]/text()')

    yield item.load_item()
```

Esto lo que va a hacer es mandar a un archivo la información cargada dentro de nuestros items.

# 2. EXTRACCION STACKOVERFLOW

**Paso 12.** No solamente podemos añadir por xpath sino que también podemos usar el método add_value que sirve para llenar una propiedad con un valor directo

Añadimos un campo nuevo id, y le asignamos un valor numérico secuencial

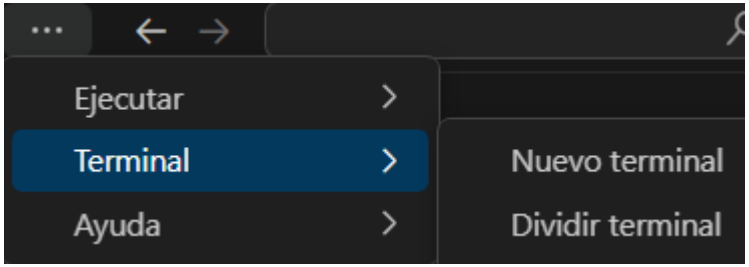
```
class Pregunta(Item):

    id = Field()

    pregunta = Field()
    descripcion = Field()
```

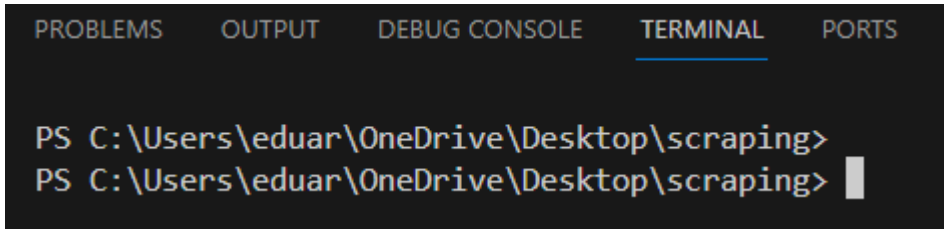
```
i=0
for pregunta in preguntas:
    item = ItemLoader(Pregunta(), pregunta)
    item.add_xpath('pregunta', './/h3/a/text()')
    item.add_xpath('descripcion', './/div[@class="s-post-summary--content-excerpt"]/text()')

    item.add_value('id', i)
    i += 1

    yield item.load_item()
```

# 2. EXTRACCION STACKOVERFLOW

**Paso 13.** Para ejecutar un script de scrapy no es tan sencillo como dar clic derecho y correr el script. Scrapy es un marco de trabajo y se trabaja desde la terminal mediante la ejecución de comandos

Dentro de Visual Studio Code, seleccionamos Nuevo Terminal. Nos ubica dentro de nuestra carpeta de proyecto



# 2. EXTRACCION STACKOVERFLOW

**Paso 14.** Tenemos que ejecutar el comando:

**scrapy runspider 4_stackoverflow.py -o resultados.csv**

**scrapy runspider <fichero_Python> -o <ficheroresults>**

Los resultados se pueden volcar en un fichero csv (de Excel, donde cada columna está separada por comas y cada fila va a representar cada uno de nuestros ítems) o en fichero json

```
PS C:\Users\eduar\OneDrive\Desktop\scraping> scrapy runspider 4_stackoverflow.py -o resultados.csv
scrapy : El término 'scrapy' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable.
Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e
inténtelo de nuevo.
En línea: 1 Carácter: 1
+ scrapy runspider 4_stackoverflow.py -o resultados.csv
+ ~~~~~~
    + CategoryInfo          : ObjectNotFound: (scrapy:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
```

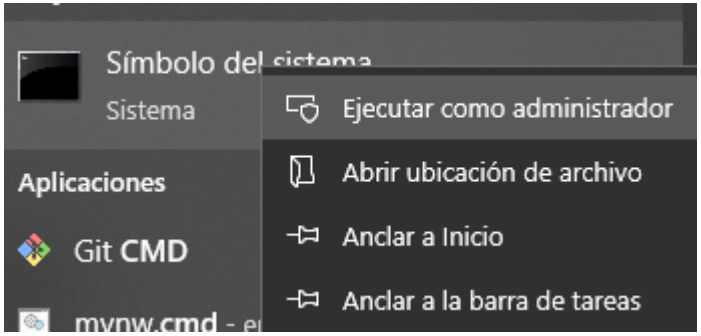
Ejecutamos y nos sale un error

# 2. EXTRACCION STACKOVERFLOW

**Paso 15.** Windows requiere que la terminal CMD donde hemos ejecutado los comandos pip de las instrucciones para instalar la librería, estén abiertas con permisos de administrador. Tenemos que ejecutar un terminal como administrator, y ejecutar:

**pip uninstall scrapy**

**pip install scrapy** (o pip3 segun sea el caso)




En caso de que utilizar Visual Studio Code, tenemos que cerrar el terminal y volverlo a abrir.

# 2. EXTRACCION STACKOVERFLOW

**Paso 16.** Ejecutamos el comando y nos muestra el feedback de todo lo que esta haciendo Scrapy: Hacer el requerimiento de la pagina al servidor, parsear la respuesta y mostrar la información extraída con xpath: id, pregunta y descripción. Al final indica que terminó bien

```
PS C:\Users\eduar\OneDrive\Desktop\scraping> scrapy runspider 4_stackoverflow.py -o resultados.csv
2024-03-17 10:12:07 [scrapy.utils.log] INFO: Scrapy 2.11.1 started (bot: scrapybot)
2024-03-17 10:12:07 [scrapy.utils.log] INFO: Versions: lxml 5.1.0.0, libxml2 2.10.3, cssselect 1.2.0, parsel 1.8.1, w3lib 2.1.
2, Twisted 24.3.0, Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)], pyOpenSSL 24.0.0 (
OpenSSL 3.2.1 30 Jan 2024), cryptography 42.0.5, Platform Windows-10-10.0.19045-SP0
2024-03-17 10:12:07 [scrapy.addons] INFO: Enabled addons:
[]
2024-03-17 10:12:07 [py.warnings] WARNING: C:\Program Files\Python312\Lib\site-packages\scrapy\utils\request.py:254: ScrapyDep
recationWarning: '2.6' is a deprecated value for the 'REQUEST_FINGERPRINTER_IMPLEMENTATION' setting.
```

```
 'response_received_count': 1,
 'scheduler/dequeued': 1,
 'scheduler/dequeued/memory': 1,
 'scheduler/enqueued': 1,
 'scheduler/enqueued/memory': 1,
 'start_time': datetime.datetime(2024, 3, 17, 9, 12, 8, 647104, tzinfo=datetime.timezone.utc)}
2024-03-17 10:12:09 [scrapy.core.engine] INFO: Spider closed (finished)
PS C:\Users\eduar\OneDrive\Desktop\scraping>
```

# 2. EXTRACCION STACKOVERFLOW

**Paso 17.** En la carpeta del proyecto se ha generado el fichero csv con los resultados de la extracción realizada

resultados.csv

```
1   descripcion,id,pregunta
2   "
3                   I'm using this (1-year-old) Nx tutorial https://www.youtube.com/watch?v=M5NwkRNrpK0
4   At 10:30 they call: ""npx nx start ngproj-a"" and everything is built just fine.
5   When I do the same thing &...
6               ",[0],The Nx build system doesn't see the projects inside the workspace
7   "
8                   Now, linux signal is handled by any thread in a process. why does not linux take a th
9               ",[1],Why does not linux kernel choice a thread oriented signal schema?
10  "
11                  I am trying to implement the custom repo with JpaRepo in spring boot 3.2.3 by followi
12  My RepoSitory Interface:
```

# 2. EXTRACCION STACKOVERFLOW

**Paso 18.** Tenemos los datos que queríamos extraer, pero no están limpios, vienen con muchos saltos de línea y/o tabulaciones. Comentaremos el campo descripción, el cual introduce mas suciedad, para que cuando generemos de nuevo el fichero csv no se genere este campo y sea mas legible

```
class Pregunta(Item):
    id = Field()
    pregunta = Field()
    #descripcion = Field()
```

```
item.add_value('id', i)
item.add_xpath('pregunta', './/h3/a/text()')
#item.add_xpath('descripcion', './/div[@class="s-post-summary--content-excerpt"]/text()')
```

# 2. EXTRACCION STACKOVERFLOW

**Paso 19.** Volvemos a ejecutar el script de scrapy y nos aparece la información mas limpia sólo con los campos id y pregunta.

resultados.csv

```
id,pregunta
[0],Filter out values that are equal on the same row?
[1],SQLite: How to include an Sqlean define statement in a `.sql` file?
[2],"Why doesn't the ( ""$@"" ) syntax work for array creation?"
[3],Why .NET Framework assemblies are not loaded inside same AppDomain of calling module?
[4],"ortools solvers GLOP, PDLP instantly writes that the model is infeasible"
[5],confuse about union and intersection type on typescript
[6],wrong map serialization - spring webflux mono
[7],How to Resolve Crashing Issue When Reading Minesweeper Level Files with High Mine Counts?
[8],Delete/modify SharePoint calendar event and sync changes with outlook calendar using MS GraphAPI
[9],Almost Everything Crash on My PC Windows 10
[10],I change the theme but the home remains the same
[11],How to remedy numerical instability in floor(t / dt) * dt != t?
[12],How to use images in Tkinter. (I'm a beginner)
[13],c++ why compiler cant find function overload when namespace is used?
[14],how to print specific list of items from a dict in python?

```

# 2. EXTRACCION STACKOVERFLOW

**Paso 20.** Lo podemos abrir con Excel, ya que se trata de un formato donde cada fila de ítem esta separada por comas. Y además tenemos un encabezado de tabla

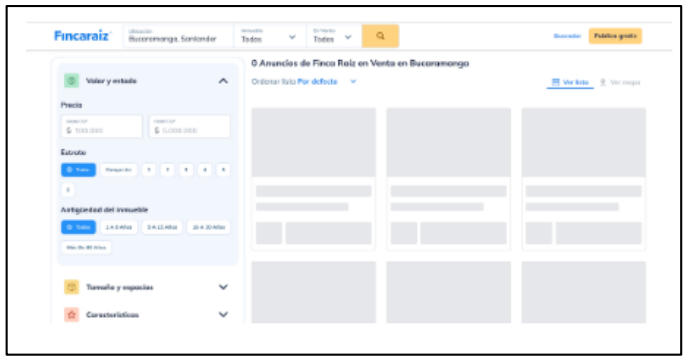
| id | pregunta | |
|---|---|---|
| [0] | Filter out values that are equal on the same row? | |
| [1] | SQLite: How to include an Sqlean define statement in a `.sql` file? | |
| [2] | "Why doesn't the ( ""$@"" ) syntax work for array creation?" | |
| [3] | Why .NET Framework assemblies are not loaded inside same AppDomain of calling module? | |
| [4] | "ortools solvers GLOP | PDLP instantly w |
| [5] | confuse about union and intersection type on typescript | |
| [6] | wrong map serialization - spring webflux mono | |
| [7] | How to Resolve Crashing Issue When Reading Minesweeper Level Files with High Mine Counts? | |
| [8] | Delete/modify SharePoint calendar event and sync changes with outlook calendar using MS GraphAPI | |
| [9] | Almost Everything Crash on My PC Windows 10 | |
| [10] | I change the theme but the home remains the same | |
| [11] | How to remedy numerical instability in floor(t / dt) * dt != t? | |
| [12] | How to use images in Tkinter. (I'm a beginner) | |
| [13] | c++ why compiler cant find function overload when namespace is used? | |
| [14] | how to print specific list of items from a dict in python? | |
| | | |

# 3. LIMITACIONES DE SCRAPY

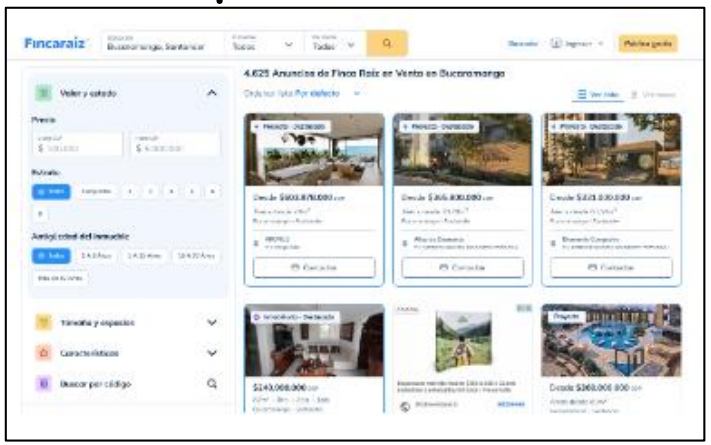
- Scrapy y requests no funcionan con todas las páginas webs. No nos sirve para extraer datos de cualquier página.
- No todas las paginas web funcionan igual
- Los diversos mecanismos que utilizan las páginas webs para cargar la información pueden afectar nuestras extracciones.
- Como por ejemplo la carga dinámica de información, que hace que lo que nos llega dentro de Requests y Scrapy sea diferente a lo que vemos en nuestro navegador.

# 3. LIMITACIONES DE SCRAPY

- Ejemplo de como se ve una página que utiliza carga dinámica de información:




- Apenas abrimos la URL vemos contenedores vacíos que unos segundos después se llenan de información



# 3. LIMITACIONES DE SCRAPY

- La carga dinámica de información también puede venir en la forma de interacciones con una página web.
- Por ejemplo, nueva información que puede aparecer al dar click en un botón que diga "CARGAR MAS" o al hacer scrolling con el mouse.