|  |  |
| --- | --- |
| Práctica 1 | WEB SCRAPING |

# Objetivo

En la presente práctica se utiliza Scrapy y Python para extraer información en formato .csv e imágenes de alojamientos publicados en la web https://www.habitaclia.com

# Contexto

La web seleccionada para realizar web scraping es [www.habitaclia.com](http://www.habitaclia.com). Concretamente se extrae la información de las diferentes páginas que aparecen en la web:

<https://www.habitaclia.com/casas-la_nora-murcia.htm>

En esta web se publican inmuebles en venta de La Ñora, Murcia. Se utiliza una araña web de Scrapy para recorrer las diferentes publicaciones de inmuebles siempre dentro del dominio [www.habitaclia.com](http://www.habitaclia.com)

# Propiedad, tecnología y ética

Utilizamos el módulo whois para conocer a quien pertenece el sitio web:

|  |
| --- |
| Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32  import whois  print(whois.whois('https://www.habitaclia.com'))  {  "domain\_name": [  "HABITACLIA.COM",  "habitaclia.com"  ],  "registrar": "Amazon Registrar, Inc.",  "whois\_server": "whois.registrar.amazon.com",  "referral\_url": null,  "updated\_date": [  "2021-12-13 06:37:07",  "2021-12-13 06:37:08.509000"  ],  "creation\_date": "2002-01-16 15:31:13",  "expiration\_date": "2023-01-16 15:31:13",  "name\_servers": [  "NS-1377.AWSDNS-44.ORG",  "NS-1863.AWSDNS-40.CO.UK",  "NS-478.AWSDNS-59.COM",  "NS-561.AWSDNS-06.NET",  "ns-1377.awsdns-44.org",  "ns-1863.awsdns-40.co.uk",  "ns-478.awsdns-59.com",  "ns-561.awsdns-06.net"  ],  "status": [  "clientTransferProhibited https://icann.org/epp#clientTransferProhibited",  "renewPeriod https://icann.org/epp#renewPeriod"  ],  "emails": [  "abuse@amazonaws.com",  "owner-964145@habitaclia.com.whoisprivacyservice.org",  "admin-964145@habitaclia.com.whoisprivacyservice.org",  "tech-964145@habitaclia.com.whoisprivacyservice.org"  ],  "dnssec": "unsigned",  "name": "On behalf of habitaclia.com owner",  "org": "Whois Privacy Service",  "address": "P.O. Box 81226",  "city": "Seattle",  "state": "WA",  "registrant\_postal\_code": "98108-1226",  "country": "US"  } |

Para conocer la tecnología aplicaremos el módulo webtech:

|  |
| --- |
| import webtech  web\_tech = webtech.WebTech()  technologies = web\_tech.start\_from\_url('https://www.habitaclia.com/', timeout=1)  print(technologies)  Target URL: https://www.habitaclia.com/  Detected technologies:  - Amazon Cloudfront  - HSTS  - Google Sign-in  - Amazon ALB  - Google Tag Manager  - Microsoft ASP.NET  Detected the following interesting custom headers:  - Server: Cloudfront  - X-Powered-By: ASP.NET  - X-Served-By: habitaclia  - X-Amz-Cf-Pop: MAD51-C1 |

Conoceremos ahora si la página web permite ser rastreada o no mediante el archivo robots.txt de la web. Para ello accedemos a:

https://www.habitaclia.com/robots.txt

|  |
| --- |
| User-agent: \*  Disallow: /hab\_usuarios/registrocorreo.asp\*  Disallow: /hab\_usuarios/ajax/\*  Disallow: /hab\_inmuebles/ajax/\*  Disallow: /dotnet/NotificacionesLiveListado/GetNotificacionesLiveListado\*  Disallow: /dotnet/solicitud/vertelefono\*  Disallow: /dotnet/solicitud/ValidarCaptcha\*  Disallow: /dotnet/ficha/favrate\*  Disallow: /dotnet/ficha/favcomment\*  Disallow: /dotnet/ficha/translate\*  Disallow: /\*.txt$  Allow: /robots.txt  Allow: /app-ads.txt  Allow: /ads.txt  Disallow: /\*ordenar=  Disallow: /\*state=  Disallow: /\*st=  Disallow: /\*geo=  Disallow: /\*lo=  Disallow: /\*filtro\_periodo=  Disallow: /\*f=  Disallow: /\*from=  Disallow: /\*filtro\_periodo=  Disallow: /\*pag=  Disallow: /\*vistamapa.htm  Disallow: /\*f\_con\_fotos=  Disallow: /\*bolIsFiltro=  Disallow: /\*tip\_op\_origen=  Disallow: /\*hUserClickFilterButton=  Disallow: /\*hMinLat=  Disallow: /\*hMinLon=  Disallow: /\*hMaxLat=  Disallow: /\*hMaxLon=  Disallow: /\*hUseLatLonFilters=  Disallow: /\*hNumPointsMapa=  Disallow: /\*list=  Disallow: /\*contactar.htm  Disallow: /q/  Disallow: /\*/q/  Disallow: /\*listainmuebles.htm  Disallow: /\*ady=  Disallow: /\*z=  Disallow: /\*fotomode=  Disallow: /\*codProv=  Disallow: /\*codPob=  Disallow: /\*openmenu=  Disallow: /\*subtipinm=  Disallow: /\*coddists=  Disallow: /\*compartirApp=  Disallow: /\*habsrc= |

En el archivo settings.py indicamos

|  |
| --- |
| ROBOTSTXT\_OBEY = True |

Indicando ‘True’ respetamos el archivo robots.txt de la web a escrapear.

Cada motor de búsqueda se identifica como un user-agent diferente

El formato de un archivo robots.txt es el siguiente:

|  |
| --- |
| Sitemap: [URL ubicación de sitemap]  User-agent: [identificador de bot]  [directiva 1]  [directiva 2]  [directiva ...]  User-agent: [otro identificador de bot]  [directiva 1]  [directiva 2]  [directiva ...] |

Donde las directivas son las reglas que los user-agents deben seguir. Hay que recordar que el robots.txt no es vinculante, es decir, indica donde no pueden acceder los rastreadores pero hay rastreadores que pueden decidir saltarse las reglas y no respetar las directivas del robotos.txt.

En el caso de habitaclia tiene un asterisco en user agent y eso indica que asigna directivas a todos los user- agents.

Como se ha indicado, el código respeta el archivo robots.txt de habiaclia al indicar

ROBOTSTXT\_OBEY = True. Tras aplicar el código se ha obtenido toda la información deseada. Es por ello que se puede decir que durante la extracción de la información no se ha violado ninguna directiva del archivo robots.txt de habitaclia.

# Descripción del dataset

El dataset extraído mediante Scrapy es totalmente accesible mediante Github a través del enlace <https://github.com/rlaborda97/Practica_1> o desde Zenodo a través del siguiente DOI <https://doi.org/10.5281/zenodo.7316460>. Se extrae la siguiente información de la web:

|  |  |  |
| --- | --- | --- |
| Número | Dato | Descripción |
| 1 | name | Nombre asignado al anuncio de la vivienda. |
| 2 | price | Precio de venta del inmueble [€]. |
| 3 | summary | Resumen de las propiedades del inmueble: superficie [m2], habitaciones, baños y precio de la superficie [€/m2]. |
| 4 | short\_description | Breve descripción del inmueble. |
| 5 | description | Descripción completa del inmueble. |
| 6 | last\_modified | Última fecha de modificación del anuncio del inmueble. |
| 7 | distribution | Distribución general del inmueble. |
| 8 | general\_characeristics | Características generales donde se detalla el año de construcción, la etiqueta de eficiencia energética, las emisiones de Co2 m2/año, aire acondicionado, si se encuentra amueblado o si incluye plaza de aparcamiento entre otros. |
| 9 | saved\_path | Ruta dentro del repositorio de Github donde se almacenan las imágenes del inmueble. |

El periodo de tiempo de extracción incluye un retraso de xx segundos para no ser detectados como rastreadores.

A continuación, se muestra la página principal donde se llevará a cabo el scraping.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

La araña web entra en cada uno de los anuncios

La araña web entra en cada uno de los anuncios y cuando llega al final de la página entrará en la siguiente página para seguir con el proceso de extracción.

Se delimita el dominio [www.habitaclia.com](http://www.habitaclia.com) para que la araña web no salga de la web.

Interfaz de usuario gráfica, Aplicación, Sitio web, PowerPoint

Descripción generada automáticamente

1

Cuando se han visitado todos los anuncios se hace click en siguiente para seguir con el proceso de extracción

A continuación mostramos el primer anuncio de la lista.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

5

4

3

2

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente

8

7

6

Se descargan todas las imágenes del anuncio.

Interfaz de usuario gráfica, Aplicación, PowerPoint

Descripción generada automáticamente

9

# Licencia

Para seleccionar la licencia nos ayudamos de la guía que GitHub proporciona en la web <https://choosealicense.com/>. Si no seleccionamos ninguna licencia estamos aceptando las leyes de derechos de autor donde no se podría reproducir, distribuir ni crear trabajos derivados.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Visitando la web descubrimos la guía “ Como elegir una licencia para su obra”

<https://www.gnu.org/licenses/license-recommendations.html>

Donde nos indican que para código de menos de 300 líneas no es necesario recurrir a licencias Copyleft pero sí recomiendan Licencia Apache 2.0. Pero esta licencia que etiquetan como blanda también permite que alguien pueda cerrar el código y que luego no se pueda distribuir libremente.

Por ello, se selecciona una licencia **GNU GPLv3** que se trata de una licencia copyleft que permite que se pueda distribuir el programa con o sin cambios pero siempre con la libertad de poder seguir haciendo más cambios y copias sin problema.

# Video

# Inspiración

Con los datos extraídos se puede calcular, entre otros, precio €/m2 de superfície, número de baños promedio, promedio de habitaciones y €/habitación.

Sí que es cierto que el precio promedio de una zona la propia web lo indica normalmente. Pero lo que se consigue con esta araña web es detectar inmuebles que han sido publicados, eliminados y vueltos a publicar. Es decir, es habitual que quien publica un inmueble lo elimine y lo vuelva a publicar de nuevo para que aparezca como novedad. Esta práctica es habitual para las empresas que publican anuncios. Por ejemplo, si un inmueble lleva publicado una semana y no se vende se suele retirar el anuncio y se vuelve a publicar como nuevo. Mediante la araña web se podrían detectar aquellos inmuebles que han sido publicados más de una vez y de esta forma conocer qué anuncio es nuevo o antiguo independientemente de la fecha de actualización. Para ello seria necesario realizar el web scraping mediante Scrapy una vez por la mañana y otra por la tarde cada día.

Al final se podría obtener un listado por antigüedad de publicación del inmueble. Esta información actualmente no la proporciona Habitaclia.

# Bibliografía

<https://www.gnu.org/licenses/copyleft.html>

<https://www.gnu.org/licenses/license-recommendations.html>

<https://choosealicense.com/>

Dimitrios Kouzis-Loukas. *Learning Scrapy. Learn the art of efficient web scaping and crawling with Python*. Packt publishing. 2016