Tipología y ciclo de vida de los datos
Aula 1

Alumnos: Enrique José Villalobos Torregrosa
Francisco Javier Albarrán González

# Práctica 1: Web scraping

## Índice:



## 1- Contexto

El presente trabajo se enmarca dentro de la asignatura "**Tipología y Ciclo de Vida de los Datos**" del Master Universitario en Ciencia de Datos de la UOC.

Dentro del capítulo dedicado al **Web scraping** se propone un ejercicio a través del cual poner en práctica los conocimientos y técnicas adquiridas.

En este contexto, tras un debate de propuestas de webs para explorar, nos pronunciamos por examinar la página web https://www.eltiempo.es/. Esta web publica semanalmente los niveles de polen de las plantas más alergénicas y que producen un deterioro en la calidad de vida de muchas personas.

Esta información se facilita a nivel de provincia, dentro del territorio español, y de tipo de planta.

Por otro lado, la web y la información que presenta son de calidad y generan la suficiente confiabilidad como para poder hacer un uso posterior de la información extraída.

Estos dos aspectos, calidad y utilidad, son los que nos decantan por esta web, a la vez que, con la información recogida a lo largo de un periodo largo de tiempo, permite el estudio de comportamiento del polen de las diferentes plantas en los diferentes territorios del país, y ver evoluciones o tendencias.

Tipología y ciclo de vida de los datos
Aula 1
Alumnos: Enrique José Villalobos Torregrosa
Francisco Javier Albarrán González

# Práctica 1: Web scraping

## 2- Título

Dada la naturaleza de la información extraída, el título no puede ser otro que "**Polen en España: distribución y seguimiento**".

La información recogida nos permite observar los diferentes pólenes en el ambiente en las fechas en las que se haya capturado y en cada provincia del territorio, y con el tiempo y sucesivas capturas de información, comprender mejor los momentos en que se producen picos de polen de algunas plantas, ver diferencias en la floración entre provincias y poder ayudar a las personas con problemas de alergia, incluso, antes de que se produzcan los picos, si conseguimos encontrar patrones de producción de polen.

## 3- Descripción del dataset

La información extraída se almacena en un csv, de manera incremental con cada día que se ejecuta el proceso, para poder obtener secuencias temporales, con la siguiente estructura:

**Día**: Campo fecha con el día de ejecución en formato DD/MM/AAAA

**Hora**: Hora de grabación de datos. Formato HH:MM:SS

**Provincia**: Nombre de provincia española

**Calidad_polen**: Nivel general de la calidad del aire. Toma el valor máximo de todos los tipos de polen extraídos. No obstante, se almacena por si la web cambiara el sentido de este dato, lo cual evitaría tener que recodificar el proceso.

**Planta**: Nombre de la planta que emite polen

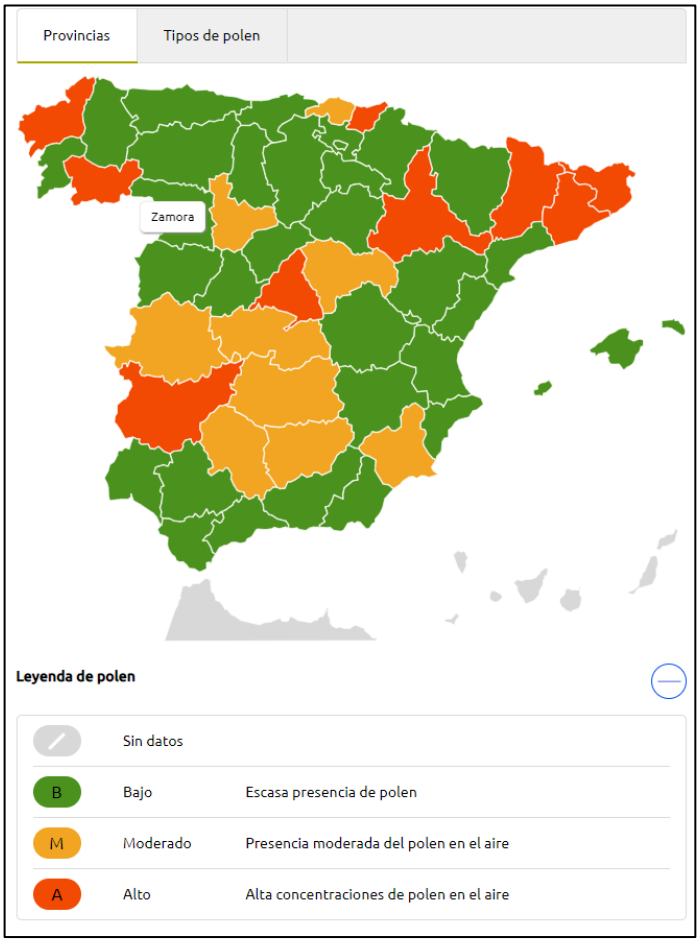
**Nivel_polen**: Campo categórico que refleja el nivel de partículas de polen en el aire. Tanto este campo como "Calidad_polen" se recogen con alguno de los siguientes valores:

- A = nivel alto de polen
- M = nivel medio
- B = nivel bajo
- En caso de no haber registro para un polen es indicativo de que no hay registro en la web para es día/provincia/polen

Alumnos: Enrique José Villalobos Torregrosa
Francisco Javier Albarrán González

# Práctica 1: Web scraping

## 4- Representación gráfica




La web representa la información a través de un "*cloropleth map*", o mapa de color como se muestra en la imagen lateral (fuente: www.eltiempo.es/polen)

Con la ejecución repetida en el tiempo se pretende poder animar la imagen o presentar la información en cronogramas temporales que nos den una idea de cómo evoluciona la producción de polen, permitiendo poner el foco en un tipo de polen en concreto o en una zona determinada.

Tipología y ciclo de vida de los datos
Aula 1
Alumnos: Enrique José Villalobos Torregrosa
Francisco Javier Albarrán González

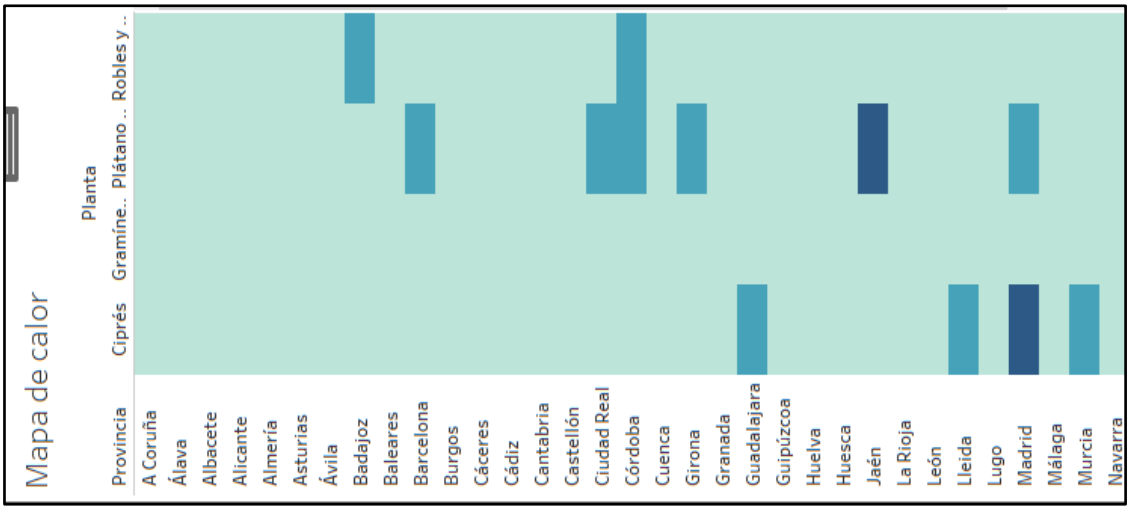
# Práctica 1: Web scraping

La información guardada queda de la siguiente manera:

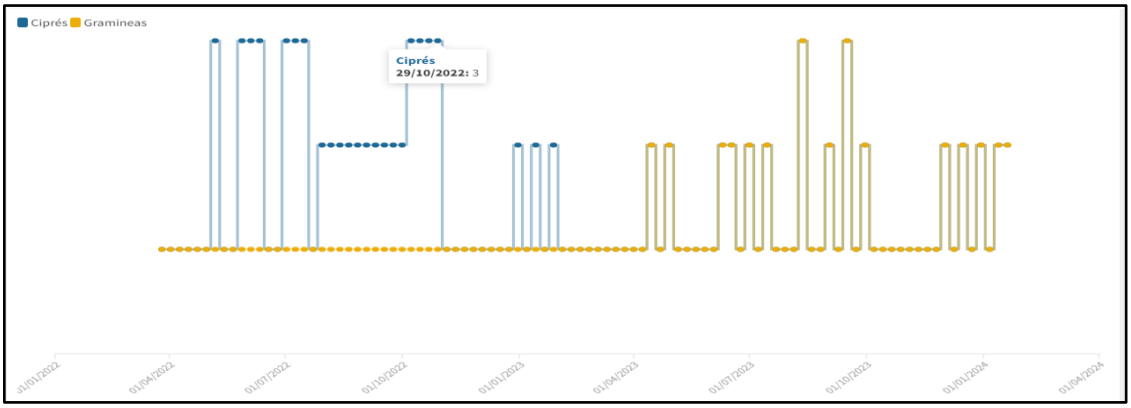
| | Dia | Hora | Provincia | Calidad_polen | Planta | Nivel_polen |
|---|---|---|---|---|---|---|
| 1 | 26/03/2022 | 21:27:11 | A Coruña | B | Ciprés | B |
| 2 | 26/03/2022 | 21:27:11 | A Coruña | B | Gramíneas | B |
| 3 | 26/03/2022 | 21:27:11 | A Coruña | B | Plátano de sombra | B |
| 4 | 26/03/2022 | 21:27:11 | A Coruña | B | Robles y encinas | B |
| 5 | 26/03/2022 | 21:27:11 | Álava | B | Ciprés | B |
| 6 | 26/03/2022 | 21:27:11 | Álava | B | Gramíneas | B |
| 7 | 26/03/2022 | 21:27:11 | Álava | B | Plátano de sombra | B |
| 8 | 26/03/2022 | 21:27:11 | Álava | B | Robles y encinas | B |
| 9 | 26/03/2022 | 21:27:11 | Albacete | B | Ciprés | B |
| 10 | 26/03/2022 | 21:27:11 | Alicante | B | Gramíneas | B |

Y con ella, una forma de representar la información, puede ser la siguiente:

**Mapa de calor**




**Cronogramas:**



Tipología y ciclo de vida de los datos
Aula 1
Alumnos: Enrique José Villalobos Torregrosa
Francisco Javier Albarrán González

# Práctica 1: Web scraping

## 5- Contenido

Se ha desarrollado programa en Python de tipo scraper que recorre la web buscando las etiquetas del código HTML que contienen la información deseada.

Existen dos niveles de profundidad. El primero para obtener los nombres de las provincias y la calidad del aire general, a nivel provincial, y un segundo, iterativo, para cada provincia, buscando los tipos de planto y su nivel de polen.

Previamente al acceso a cada nivel se busca un código dinámico tras el cual se encuentran las url de cada página a la que se accede, tanto para las provincias como para los tipos de planta.

La información se almacena en un dataset que, al finalizar el proceso, se graba en formato csv estándar en la carpeta "src" del proyecto.

## 6- Agradecimientos

En primer lugar, queremos agradecer a la web la información y el servicio que prestan.

```
User-agent: *
Disallow:
```

En cuanto al procedimiento, se ha consultado el fichero "robots.txt" que no impide que procesos como el diseñado accedan a la web de manera automática, a pesar de tener un servicio de suscripción.

```
In [8]: AGENT_NAME = 'polen'
URL_BASE = 'https://eltiempo.es/'
parser = urllib.robotparser.RobotFileParser()
parser.set_url(parse.urljoin(URL_BASE, 'robots.txt'))
parser.read()

PATHS = [
    '/',
    '/polen']

for path in PATHS:
    print('{!r:>6} : {}'.format(
        parser.can_fetch(AGENT_NAME, path), path))
    url = parse.urljoin(URL_BASE, path)
    print('{!r:>6} : {}'.format(
        parser.can_fetch(AGENT_NAME, url), url))
    print()
```

```
  True : /
  True : https://eltiempo.es/

  True : /polen
  True : https://eltiempo.es/polen
```

Asimismo, se han examinado las cláusulas de "Aviso legal" con las condiciones generales para el acceso, no encontrando impedimentos para el automatismo diseñado. Entre el clausulado se indica "*El Usuario se compromete a no trasmitir, introducir, difundir y poner a disposición de terceros, cualquier tipo de material e información (datos contenidos, mensajes, dibujos, archivos*

Alumnos: Enrique José Villalobos Torregrosa
Francisco Javier Albarrán González

# Práctica 1: Web scraping

*de sonido e imagen, fotografías, software, etc.) que sean contrarios a la ley, la moral, el orden público y los presentes Términos de Uso y, en su caso, a las Condiciones Particulares que le sean de aplicación*", cosa que cumplimos. Entendemos, por tanto, que esta cláusula no se encuentra violentada ya que no atentamos contra la ley ni el orden público con la actividad realizada, además de no percibir ningún tipo de lucro por ello ni por la información extraída.

**La finalidad de la extracción vía scraper de la información de la web eltiempo.es es meramente académica**.

Otras revisiones realizadas antes de iniciar el proyecto, además de "robots.txt" han sido comprobar la tecnología o el propietario de la web por si hubiera alguna información adicional que fuera relevante a la hora de decidirnos por esta web.

```
In [11]: import builtwith
         print(builtwith.builtwith("https://www.eltiempo.es/"))

         {'web-servers': ['OpenResty', 'Nginx'], 'programming-languages': ['Lua'], 'advertising-networks': ['AppNexus', 'Rubicon Projec
         t'], 'font-scripts': ['Google Font API'], 'tag-managers': ['Google Tag Manager'], 'web-frameworks': ['Twitter Bootstrap']}

In [10]: import whois
         print (whois.whois("https://www.eltiempo.es/"))

         {
           "domain_name": null,
           "registrar": null,
           "whois_server": null,
           "referral_url": null,
           "updated_date": null,
           "creation_date": null,
           "expiration_date": null,
           "name_servers": null,
           "status": null,
           "emails": null,
           "dnssec": null,
           "name": null,
           "org": null,
           "address": null,
           "city": null,
           "state": null,
           "zipcode": null,
           "country": null
         }
```

## 7- Inspiración

Una de motivaciones para elegir esta web fue la de utilidad. El motivo de que la información extraída pueda ser relevante para otras personas, biólogos, estadistas o, sencillamente, personas que sufren de alergía es suficiente como para desarrollar el proceso que automatice la extracción de esta información. El proyecto se puede enriquecer con información meteorológica del día en que se extraen los datos y así poder mejorar las previsiones o entender mejor el comportamiento de ciertos pólenes.

Nos hubiera gustado poder alcanzar a las mediciones del molen, en µgr/$m^3$ de aire, pero esta información no estaba disponible. En todo caso es una buena aproximación que esperamos que pueda ayudar a estar personas.

Alumnos: Enrique José Villalobos Torregrosa
Francisco Javier Albarrán González

# Práctica 1: Web scraping

Los usos, como ya se ha mencionado, pueden ir desde la mera consulta (para la cual ya está la propia web), a tener un histórico de la evolución de la floración de determinadas especies de plantas que producen alergias y poder estudiar los momentos en que ésta se produce, los picos de máxima actividad, y anticiparnos a las medidas a tomar entre la población afectada, ya sea mediante alertas tempranas, uso de mascarilla o aprovisionar antihistamínicos.

## 8- Licencia

La licencia para el dataset y el programa desarrollado es CC0 1.0 Universal porque consideramos que, quien quiera, puede tener acceso a este conjunto de datos y a continuar el proyecto, ya sea enriqueciéndolo con los datos que considere oportuno o con las ejecuciones mantenidas en el tiempo para los fines que considere.

## 9- Código

El programa se ha programado íntegramente en Python utilizando, para ello, librerías como "Pandas", para el tratamiento de datasets, "requests" y "BeautyfullSoup4" para el manejo de HTML y su interpretación y manejo automático, o "csv" para proceder a la grabación en fichero csv de la información extraída.

El código, así como el dataset generado, se encuentra en el repositorio GitHub siguiente:

https://github.com/JAlbarrn/TCV_P1_WebScraping

## 10- Dataset

Se ha subido a Zendolo el dataset originado (versión 1.1 – datos a 4 de abril de 2022) con el DOI 10.5281/zenodo.6413017.

## 11- Vídeo

Como complemento, el presente proyecto cuenta con un vídeo explicativo de las funciones desarrolladas, tanto del código como de algunos problemas encontrados en el diseño y construcción del proyecto.

| Contribuciones | Firma |
|---|---|
| Investigación previa | EJVT, FJAG |
| Redacción de las respuestas | EJVT, FJAG |
| Desarrollo del código | EJVT, FJAG |