

# Film Scraping

TIPOLOGÍA Y CICLO DE VIDA DE LOS DATOS

JORGE PUENTE DUARTE

JIMMY ACOSTA SUAREZ

NOVIEMBRE DE 2020

# Tabla de Contenido

- Contexto ..... 2
- Definición de la base de datos ..... 3
- Procedimiento para obtener los datos ..... 5
  - Creación de funciones ..... 7
- Descripción de FilmHistTable..... 9

## Contexto

La industria del entretenimiento audiovisual no para de crecer, acorde al Theme Report del 2019<sup>1</sup> de emitido por Motion Picture association, para el 2019 representó un mercado de 100 billones de dólares. Con el crecimiento de la industria, se sobre entiende que existe un constante crecimiento de la oferta y la demanda, lo que quiere decir que hay un crecimiento constante en el número de estrenos de cada año, existen cientos de películas nuevas cada año, (según el reporte, 835 para el 2019, casi 100 más comparado con el año anterior). Son muchas personas a quienes les gusta disfrutar de una gran película, sin embargo, en los tiempos actuales, en los cuales el tiempo es un bien preciado y que para ver al menos un 10 por ciento de las películas disponibles cada año, una persona tendría que dedicar al menos 150 horas.

Acorde al reporte, las personas entre 25 y 39 años representan la mayor población que asiste al cine, con un promedio de 4.2 asistencias por año al cine por persona. Lo que representa que cada persona que asiste ocasionalmente al cine espera seleccionar la película correcta, aquella película que cumpla con sus gustos. Existen páginas de críticas de cine con calificaciones y revisiones, sin embargo, puede tomar tiempo hasta que muchas personas hagan su aporte para aquellos que se guían de la calificación puedan tener un criterio adecuado para seleccionar una película que mejor se ajuste a sus gustos.

Teniendo en cuenta estos factures, sería de utilidad obtener un dataset con películas que contenga información necesaria para poder predecir gustos basados en apreciaciones anteriores, títulos, géneros, directores, efectos visuales, calificaciones, etc.

En este sentido, hemos identificado una página web que nos puede ayudar a cumplir este cometido, pretendemos hacer scraping de la página web de Filmaffinity con el fin de tratar sus datos y poder realizar diferentes análisis posteriores:

<https://www.filmaffinity.com/es/advsearch.php>

está página web, es un portal en el que lo aficionados al cine tienen la oportunidad de crear su usuario, calificar películas, dejar su crítica y compartir impresiones. Esta misma dispone de un listado histórico de películas, documentales y series de televisión que puede ser consultado basado en criterios como país, año, título, género.

El resultado de cada búsqueda es mostrado en un listado de 20 películas por página, con la posibilidad de continuar en las siguientes páginas con el fin de identificar todos los resultados.

En el presente documento explicaremos como fueron obtenidos los datos, la estructura de dicha página y la descripción del los datos extraídos a través de la aplicación de técnicas de Web Scraping usando Python.

---

<sup>1</sup> El reporte puede ser consultado en <https://www.motionpictures.org/wp-content/uploads/2020/03/MPA-THEME-2019.pdf>

# Definición de la base de datos

Hemos definido llamar el dataset como *FilmHistTable* la cual corresponde a la extracción de datos de la página de filmaffinity.

## Identificación del archivo Robots.txt

A partir de la identificación del archivo ubicado en <https://www.filmaffinity.com/robots.txt>, obtenemos los siguientes resultados:

```
User-agent: *  
Disallow: /*?FASID  
Disallow: /*&FASID  
Disallow: /*/sharerating  
Disallow: /flash/rats.swf
```

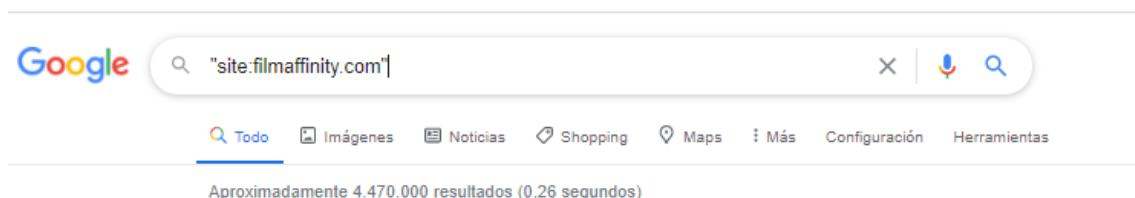
De este archivo, podemos determinar que el autor da permiso completo a todos los robots, sin embargo, no permite tener acceso a tres directorios datos dentro FASID y damos dentro sharerating.

## Identificación del mapa del sitio

Luego de realizar varias búsquedas, no fue identificado ningún mapa del sitio

## Tamaño del sitio

A partir de una búsqueda inicial del sitio, obtenemos que tiene más de cuatro millones de resultados, lo que representa que tendremos que hacer diferentes iteraciones de consultas para obtener los datos.



## Propietario

Obtenemos la información del propietario usando el siguiente código:

```
pip install python-whois  
  
import whois  
  
print(whois.whois('https://www.filmaffinity.com/'))
```

Para lo cual obtenemos:

```
{
  "domain_name": [
    "FILMAFFINITY.COM",
    "filmaffinity.com"
  ],
  "registrar": "Arsys Internet, S.L. dba NICLINE.COM",
  "whois_server": "whois.nicline.com",
  "referral_url": null,
  "updated_date": [
    "2020-06-21 07:21:16",
    "2015-01-13 15:24:07"
  ],
  "creation_date": "2001-06-20 14:23:27",
  "expiration_date": "2021-06-20 14:23:27",
  "name_servers": [
    "NS1.FILMAFFINITY.COM",
    "NS2.FILMAFFINITY.COM"
  ],
  "status": [
    "ok https://icann.org/epp#ok",
    "ok https://www.icann.org/epp#ok"
  ],
  "emails": [
    "email@nicline.com",
    "whoiscontact@domainconnection.info"
  ],
  "dnssec": [
    "unsigned",
    "Unsigned"
  ],
  "name": "REDACTED FOR PRIVACY",
  "org": null,
  "address": "REDACTED FOR PRIVACY",
  "city": "REDACTED FOR PRIVACY",
  "state": "Madrid",
  "zipcode": "REDACTED FOR PRIVACY",
}
```

```
"country": "ES"  
}
```

De la información anterior podemos ver que el dominio desde el que queremos hacer las consultas pertenece a `NICLINE.COM`, y el mismo está activo hasta el 2021. Al hacer consultas, no vemos inicialmente ningún riesgo para llevar a cabo el proceso de Scraping.

## Procedimiento para obtener los datos

Este ejercicio fue realizado a partir uno de los recursos con el que cuenta Filmaffinity para dar a los usuarios la posibilidad de realizar búsquedas dentro del listado completo de datos a partir de diferentes criterios de búsqueda:

<https://www.filmaffinity.com/es/advsearch.php>

En el momento de ejecutar una petición de búsqueda usando algunos criterios específicos, la página responde a partir de un URL que contiene los criterios de búsqueda definidos como se muestra en la siguiente URL:

[https://www.filmaffinity.com/es/advsearch.php?page=1&stype\[\]=title&country=JP&fromyear=2016&toyear=2016](https://www.filmaffinity.com/es/advsearch.php?page=1&stype[]=title&country=JP&fromyear=2016&toyear=2016)

Como se muestra en el anterior ejemplo, está petición contiene el resultado para los siguientes criterios de búsqueda:


Country=JP (Japón)

Fromyear:2016

Toyear:2016

Lo que quiere decir que estamos buscando todos los datos con fecha de lanzamiento de 2016 y cuyo país de origen sea Japón.

Para este ejemplo como lo observamos en la siguiente imagen, estamos obteniendo un total de 25 páginas (máximo número de páginas por cada consulta), cada una con 20 títulos por página, (el máximo número mostrado por página), en la URL podemos observar que el criterio de número de página está definido por *"Page"* en este caso *"Page=1"*. Otros criterios pueden ser introducidos, por ejemplo, el género.



**Usuarios**  
Votar los tours  
Iniciar sesión  
Registrarse

**España**  
Películas en cartelera  
Cines España  
Próximos estrenos  
Estrenos Blu-ray venta  
Próximos Blu-ray venta  
Ya para alquilar  
Próximamente alquilar  
Video on Demand  
Netflix  
Netflix (próx)  
Movistar +  
Movistar + (próx)  
HBO  
Filmin  
Rakuten TV  
Amazon Prime  
Disney+  
Apple TV+

**USA - UK - FR**  
Estrenos USA  
Estrenos Reino Unido  
Estrenos Francia

**Secciones**

**BUSCADOR AVANZADO** (Buscará en los campos que selecciones)  

Texto:

☒ Título
☐ Director
☐ Reparto
☐ Guión  
☐ Fotografía
☐ Música
☐ Productora

País:  Género:

Año: desde  hasta


**BUSCADOR GLOBAL** (Buscará en todas las fichas de películas y críticas)  

Texto:

Búsqueda limitada a 500 resultados. Por favor sea más específico.

Página 1 de 25
500 resultados

>>


Your Name (2016)

Una vez identificado la herramienta disponible dentro del URL para explorar el contenido de la base de datos completa, se ha creado una serie de bucles “for” anidados para iterar año por año y género por género, para ello inicialmente hemos creado un pequeño diccionario que contiene los géneros disponibles dentro de las alternativas de búsqueda:

Code	Género
'AC'	'Acción'
'AN'	'Animación'
'AV'	'Aventuras'
'C-F'	'Cienciaficción'
'DR'	'Drama'
'FAN'	'Fantástico'
'INF'	'Infantil'

Code	Género
'INT'	'Intriga'
'TE'	'Terror'
'WE'	'Western'
'CO'	'Comedia'
'DO'	'Documental'
'RO'	'Romance'
'TH'	'Thriller'
'WE'	'Western'

Como rango de años, hemos definido entre 1900 y 2020, en este caso, teniendo en cuenta la cantidad de géneros incluidos en el diccionario y la cantidad de años, esto nos da un total de 121 años y 15 géneros, lo cual representa un total de 1,815 consultas, para cada consulta, puede existir un largo número de resultados, en el caso que el número de resultados exceda un valor de 20, creará una nueva página, con ese número máximo de títulos por página, por lo cual, hemos creado otro bucle para realizar la consulta por cada página siguiente.

Nota: teniendo en cuenta la cantidad de peticiones, hemos decidido para el ejercicio, limitar la cantidad de páginas a consultar a un total de 15 páginas. Esta situación ocurre especialmente en años próximos al presente (pues a inicios del siglo XX la densidad de títulos es claramente inferior).

Cabe destacar que ha sido necesario introducir “delays” entre iteraciones usando la función *time.sleep*, debido a que durante los diferentes ejercicios, fuimos expulsados de la página, registrando una respuesta “<Response [429]>”.

Hemos creado funciones separadas para la extracción de cada uno de los atributos de interés aparte de los dos atributos iniciales que usamos para realizar las consultas, como lo es el año y el género.

## Creación de funciones

Creamos un archivo con las funciones que se encargarán de extraer los datos para cada uno de los atributos, para ello, hemos creado un total de 5 funciones:

- movie\_titles
- movie\_countries
- movie\_marks
- movie\_direction
- movie\_casting



Por medio de estas, en cada una de las solicitudes iniciales, son seleccionados los elementos que corresponden dentro del contenido HTML a cada uno de los atributos específicos de cada request, por medio del uso de la función *BeautifulSoup* Parseamos el contenido y lo asignamos a una variable Soup, desde la cual cada una de las funciones, realizarán la extracción del contenido que requerimos para llenar nuestro dataset.

Al crear listas vacías para cada uno de los atributos, con cada una de las consultas realizamos una agregación, de esta manera vamos obteniendo los resultados acumulados.

Existen títulos que, dentro del atributo de director o casting, puede contener varios valores, para este caso, fue necesario hacer un *for* adicional con el fin de identificar los mismos y en el caso de existir mas de uno, separarlos y asignarlos a una lista temporal, para luego asignar los valores totales contenidos a la lista que contiene todos los atributos del dataset.

Posteriormente hemos renombrado los nombres de los atributos, y para el caso de casting y director, teniendo en cuenta que fue creada una sublista con los diferentes componentes, hemos hecho un join para dejar todos los valores en una misma columna separada por “,”.

Posteriormente, hemos creado un nuevo atributo con el nombre de “Tipo Filme” en el cual rellenamos para obtener información contenida dentro del título de las películas, una clasificación adicional: (**C**): Cortometraje, (**TV**): Estreno televisivo, (**Serie de TV**): Serie, (**Documental**): Documental, (**Miniserie de TV**): Miniserie y en el caso contrario es considerada una película.

Como último paso, hemos identificado que un título puede estar clasificado dentro de varios géneros, teniendo en cuenta que en el planteamiento inicial, hemos hecho la consulta por género y año, esto nos trae como consecuencia que un título estará duplicado dentro de nuestro dataset, tantas veces como géneros tenga asociados, por lo mismo, hemos realizado una de-duplicación de los títulos, agrupando el listado completo por los demás campos y creando un listado dentro del frame de Genero, para luego hacer un Join de estos valores y poner todos los valores dentro del mismo campo separados por “,”.

## Descripción de filmaffinity\_dataset

Este data set contiene un listado de películas, documentales, series de televisión entre otros, para los años comprendidos entre 1900 y 2020, a continuación, describimos cada uno de los campos que componen esta tabla:

- Título: String que contiene el título de cada película, en algunas ocasiones dentro del título puede estar incluida la categoría, como el caso de series de televisión (Serie de TV) o Cortometraje (C), información que hemos usado para crear otro parámetro. (Tipo filme)
- País: String que contiene el país de origen de cada película o título.
- Año: Numérico, Año de lanzamiento.
- Género: String, clasificación del género de cada película: puede estar entre: Acción, Animación, Aventuras, Bélico, Ciencia Ficción, Cine Netro, Comedia, Desconocido, Documental, Drama, Fantástico, Infantil, Intriga, Musical, Romance, Serie de TV, Terror, Thriller, Western.
- Dirección: String que contiene los nombres de los directores de la película.
- Reparto: String con el listado de actores que participaron en la película.
- Tipo filme: String, variable obtenida a partir del título, en el cual puede estar entre: "Cortometraje", "Documental", "Estreno televisivo", "Miniserie", "Película" o "Serie".
- Nota: Numeric, float, campo que describe la nota promedio que ha recibido cada elemento según las evaluaciones previas de los usuarios. Valor que está entre 0 y 10.