



# Recuperación de la Información y Web Semántica

## **Autores**

*Joaquín Solla Vázquez*

*David Zambrana Seoane*

# Índice

Elección de la temática.....	4
Tecnologías usadas.....	5
Crawling, manejo y almacenamiento de la información.....	5
Desarrollo web.....	5
Herramientas auxiliares.....	5
El <i>crawler</i> .....	6
Adaptación de los datos y volcado a <i>ElasticSearch</i> .....	8
La web.....	9
Filtros utilizados.....	10
Barra de búsqueda y lista de filtros activos.....	11
Lista de resultados.....	12
Ejecución y acceso al proyecto.....	13
Ficheros y carpetas relevantes.....	13
Ejecución del proyecto.....	13
Acceso al proyecto.....	13
Conclusiones y trabajo futuro.....	14

## Índice de figuras

Figura 1: Vista a la página web de IMDb.....	4
Figura 2: Urls configuradas para el spider.....	6
Figura 3: Campo title como text.....	8
Figura 4: Campo title como keyword.....	8
Figura 5: Vista inicial de la web.....	9
Figura 6: Configuración de seguridad para Elasticsearch.....	9
Figura 7: Configuración de CORS para Elasticsearch.....	9
Figura 8: Filtros de búsqueda de la web.....	10
Figura 9: Autocompletado de la barra de búsqueda.....	11
Figura 10: Resultados de una búsqueda con un error tipográfico.....	11
Figura 11: Lista de filtros activos.....	11
Figura 12: Información de una película con sus detalles técnicos.....	12
Figura 13: Lista de métodos de ordenación.....	12

# Elección de la temática

Para el desarrollo de este proyecto se ha optado por elegir la temática de las películas, la cual tiene un público muy grande, activo y cuenta con gran número de webs e información verificada, la cual resulta muy útil para la obtención de información.

De forma específica, el propósito del proyecto consta de 3 fases:

1. *Crawling*: Obtención de información acerca de un gran número de películas (título, año, idioma original, reparto principal, póster, etc.) mediante la técnica de *crawling* a una página web dedicada a la información sobre películas.
2. Procesamiento de la información: Una vez obtenidos los datos de las películas, estos deben filtrarse, eliminarse errores y formatearse de la forma correcta. Después de esto se han de copiar a una base de datos para que sean mapeados y preparados para su consulta.
3. Creación de una web: Con la base de datos preparada, se desarrollará una página web que se comunique con ella y pueda obtener los datos de las películas almacenadas. Además, han de aplicarse diferentes tipos de filtros a la búsqueda (por palabras, ordenación, por campos numéricos, campos múltiples, etc.) de forma que el usuario de la web pueda encontrar lo que está buscando de entre todos los resultados.

De esta forma, para la primera fase se ha elegido una página web (con solamente una ya bastaba para obtener suficientes resultados) de la que obtener la información. Esta ha sido la web oficial de IMDb (<https://imdb.com/>), una de las organizaciones más prestigiosas y reputadas en el mundo del cine, su información y reseñas, lo cual encaja perfectamente con lo que estábamos buscando. Además, cuenta con un catálogo de películas prácticamente completo.



Figura 1: Vista a la página web de IMDb.

# Tecnologías usadas

Las tecnologías que se han utilizado a lo largo del proyecto se pueden dividir en 3 grupos diferentes:

## ***Crawling, manejo y almacenamiento de la información***

**Python:** El lenguaje de programación empleado para el desarrollo del *crawler* y su *spider*, de forma que todo el código para la obtención de la información de la web, que se podría entender como el *backend* de la aplicación, es *Python 3*.

**Scrapy:** Un *framework* dedicado a la obtención de información de páginas webs mediante *bots* o *spiders*, los cuales van de enlace en enlace obteniendo los campos indicados y almacenándolos en nuestro caso en un archivo *JSON*.

**JSON:** Formato de texto en el que se han almacenado los datos obtenidos tras el *crawling* para su posterior procesamiento y almacenamiento en la base de datos.

**ElasticSearch:** Base de datos o motor de búsqueda utilizado para el almacenamiento y posterior lectura de la información. Mediante su ejecución en el puerto 9200, la página web puede comunicarse con ella para recuperar las películas que el usuario está buscando.

**Postman:** Plataforma utilizada para el lanzamiento de peticiones *HTTP* para la correcta configuración y mapeo de los datos dentro de *ElasticSearch* de forma que se gestiona la creación de índices, el mapeo de campos, el volcado de información y más.

## ***Desarrollo web***

**React:** Una de las librerías más conocidas para el desarrollo de *frontend*.

**ReactiveSearch:** Una librería dedicada a enlazar *React* y *ElasticSearch* que además brinda un catálogo de filtros y elementos ya creados para la búsqueda de información con estas dos tecnologías.

**Antd:** Librería que proporciona objetos ya creados como filas y columnas para acelerar y simplificar el desarrollo de las interfaces de usuario.

**DOMPurify:** Librería para *React* dedicada a limpiar y formatear los datos de forma correcta para evitar problemas de formatos y *display* con las webs *React*.

## ***Herramientas auxiliares***

**Git:** Sistema de control de versiones utilizado.

**GitHub:** Portal de control de versiones en el que se ha creado el repositorio del proyecto y, por lo tanto, se ha subido todo su código.

**Multi ElasticSearch Head:** Extensión para navegadores web para consultar información del estado del servidor de *ElasticSearch* y ejecutar peticiones de forma rápida y sencilla.

## El crawler

Para el *crawling*, se ha definido un *spider* configurado para que tenga una espera de 3 segundos entre cada ítem leído (para evitar posibles vetos) y que no tenga límite de ítems, de forma que leerá todas las películas disponibles en las *urls* proporcionadas. El *spider* obedece a los archivos *ROBOTS.txt* por lo que no se ha realizado ninguna actividad “no deseada” para la web que nos ha proporcionado la información.

El dominio que se ha utilizado es el de la web raíz ([www.imdb.com](http://www.imdb.com)) y posteriormente se ha definido una lista con todas las *urls* que el *bot* ha de visitar dentro de esa web:

```
class FilmsCrawlerSpider(CrawlSpider):
    name = "no-scraper"
    allowed_domains = [
        'www.imdb.com'
    ]
    start_urls = [
        "https://www.imdb.com/chart/top/?ref_=nv_mv_250",
        "https://www.imdb.com/chart/moviemeter/",
        "https://www.imdb.com/search/title/?title_type=feature&genres=action",
        "https://www.imdb.com/search/title/?title_type=feature&genres=adventure",
        "https://www.imdb.com/search/title/?title_type=feature&genres=animation",
        "https://www.imdb.com/search/title/?title_type=feature&genres=biography",
        "https://www.imdb.com/search/title/?title_type=feature&genres=comedy",
        "https://www.imdb.com/search/title/?title_type=feature&genres=crime",
        "https://www.imdb.com/search/title/?title_type=feature&genres=documentary",
        "https://www.imdb.com/search/title/?title_type=feature&genres=drama",
        "https://www.imdb.com/search/title/?title_type=feature&genres=family",
        "https://www.imdb.com/search/title/?title_type=feature&genres=fantasy",
        "https://www.imdb.com/search/title/?title_type=feature&genres=film-noir",
        "https://www.imdb.com/search/title/?title_type=feature&genres=history",
        "https://www.imdb.com/search/title/?title_type=feature&genres=horror",
        "https://www.imdb.com/search/title/?title_type=feature&genres=music",
        "https://www.imdb.com/search/title/?title_type=feature&genres=musical",
        "https://www.imdb.com/search/title/?title_type=feature&genres=mystery",
        "https://www.imdb.com/search/title/?title_type=feature&genres=romance",
        "https://www.imdb.com/search/title/?title_type=feature&genres=sci-fi",
        "https://www.imdb.com/search/title/?title_type=feature&genres=sport",
        "https://www.imdb.com/search/title/?title_type=feature&genres=thriller",
        "https://www.imdb.com/search/title/?title_type=feature&genres=war",
        "https://www.imdb.com/search/title/?title_type=feature&genres=western"
    ]
]
```

Figura 2: Urls configuradas para el spider.

Estas *urls* corresponden con tops de películas definidos por IMDb, como el top mejores 250 películas, películas más populares en la actualidad o las mejores 50 películas por género (de aquí que, al ser tops con una extensión definida y no muy extensa, se ha eliminado el límite de ítems al *spider*).

Como *Scrapy* gestiona las *urls* ya visitadas, si una película aparece en varios tops diferentes, esta solo será almacenada una vez, de esta forma se evitan duplicados.

Se ha definido una regla para que el *spider* entre en todos los elementos cuya *url* sea la de IMDb precedida de “/title/tt”, que en este dominio corresponde con la página de detalles de todas las películas.

Para la obtención de algunos campos como fechas, la duración o la valoración de la película, se han creado funciones auxiliares que se encargan de formatear y procesar estos datos en la forma deseada para nuestro proyecto.

Los campos almacenados de cada película son 14:

- Título (title)
- Fecha de lanzamiento (release\_date)
- Argumento (brief\_plot)
- Reparto principal (popular\_cast)
- Director (director)
- Guionistas (scriptwriter)
- Duración en minutos (duration)
- Producción (production)
- País de origen (original\_country)
- Idioma original (original\_language)
- Guía parental (parental\_guide)
- Valoración (score)
- Género (genre)
- *Url* del póster (poster\_url)

Este es un ejemplo de la información obtenida para una película:

**"title":** "The Godfather",

**"release\_date":** "1972-03-24",

**"brief\_plot":** "Don Vito Corleone, head of a mafia family, decides to hand over his empire to his youngest son Michael. However, his decision unintentionally puts the lives of his loved ones in grave danger...",

**"popular\_cast":** ["Marlon Brando", "Al Pacino", "James Caan", "Diane Keaton", "Richard S. Castellano", "Robert Duvall", "Sterling Hayden", "John Marley", "Richard Conte", "Al Lettieri", "Abe Vigoda", "Talia Shire", "Gianni Russo", "John Cazale", "Rudy Bond", "Al Martino", "Morgana King", "Lenny Montana"],

**"director":** ["Francis Ford Coppola"],

**"scriptwriter":** [],

**"duration":** 175,

**"production":** ["Paramount Pictures", "Albert S. Ruddy Productions", "Alfran Productions"],

**"original\_country":** "United States",

**"original\_language":** "English",

**"parental\_guide":** "R",

**"score":** 4.6,

**"genre":** "Crime",

**"poster\_url":**

"https://m.media-amazon.com/images/M/MV5BM2MyNjYxNmUtYTAwNi00MTYxLWJmNWYtYzZlODY3ZTk3OTFIXkEyXkFqcGdeQXVyNzkwMjQ5NzM@.\_V1\_FMjpg\_UX1000\_.jpg"

# Adaptación de los datos y volcado a *ElasticSearch*

Cuando el *crawler* ha terminado de obtener todos los datos (que en este caso lleva más de una hora debido a las pausas de 3 segundos), el siguiente paso es adaptar la información del archivo *JSON* obtenido para que pueda ser volcado a *ElasticSearch* correctamente. Esto se lleva a cabo con la ejecución del archivo *postman\_formatter.py*, ubicado dentro del proyecto del *crawler*. La ejecución de este archivo tiene 4 fases:

1. Comprobación de duplicados.
2. Eliminación de ítems con errores de formato.
3. Agregación de los campos `{"create": {"_id": X}}`.
4. Formato final y generación del *JSON* para el volcado a *ElasticSearch*.

El resultado final de este código es un archivo llamado *films\_formatted.json* que contiene todas las películas con el formato necesario para el volcado en *ElasticSearch* a través de *Postman*.

Con la información formateada, pasamos a la creación del índice a través de *Postman*, que se llama en este caso *scrapyfilms\_index*.

En el mapeo de los campos del índice, se ha de definir bien qué tipo de dato es cada columna. Un buen ejemplo de esto es el campo *title*, por el cual se necesita realizar una búsqueda por palabras (de forma que debe ser de tipo *text*) y a su vez también se quieren ordenar los datos (por lo que debería ser de tipo *keyword*). La solución a este problema ha sido definir una columna *title* del tipo *text* para la búsqueda por palabras que a su vez tiene el campo *copy\_to a title\_keyword*, que es el mismo campo pero definido como *keyword*, de esta forma tenemos el título de ambas formas y podemos realizar ambas operaciones sobre él

```
"title": {  
  "type": "text",  
  "copy_to": "title_keyword"  
},
```

Figura 3: Campo *title* como *text*.

```
"title_keyword": {  
  "type": "keyword"  
}
```

Figura 4: Campo *title* como *keyword*.

De forma resumida, los campos que se van a utilizar para la búsqueda por palabras se han definido de tipo *text*, la fecha como *date* (con el formato “yyyy-MM-dd” para asegurar el correcto funcionamiento con el filtro de intervalo de fechas), los campos numéricos como *integer* y *float* según corresponda y aquellos campos que son texto pero sobre los que se filtrará (como el idioma o el país de origen) son definidos como *keyword*.

Posteriormente se vuelcan los datos en *ElasticSearch* enviando todo el contenido del archivo *films\_formatted.json* a través de una petición *PUT*.

Una vez hechos estos pasos, los datos están listos para ser consultados. En la versión actual del proyecto, el motor de búsqueda cuenta con **730 películas** en total.

Todas las peticiones *HTTP* realizadas en *Postman* están disponibles en el proyecto en formato *JSON* en el archivo *scrapyfilms.postman\_collection.json*.



# La web

El diseño de la página web se ha simplificado gracias a la librería *antd*. Consta de dos columnas principales: una para los filtros y otra para la barra de búsqueda y la lista de resultados.

Con el paquete *ReactiveSearch* se ha acelerado significativamente el desarrollo de la interfaz gracias a los filtros ya construidos que proporciona. El componente principal de estos es *ReactiveBase*, que configura la conexión a la dirección y al índice deseados de *ElasticSearch*, dentro de este componente se encuentran todos los demás como por ejemplo *MultiLists*, *DataSearch*, *RangeInput*...

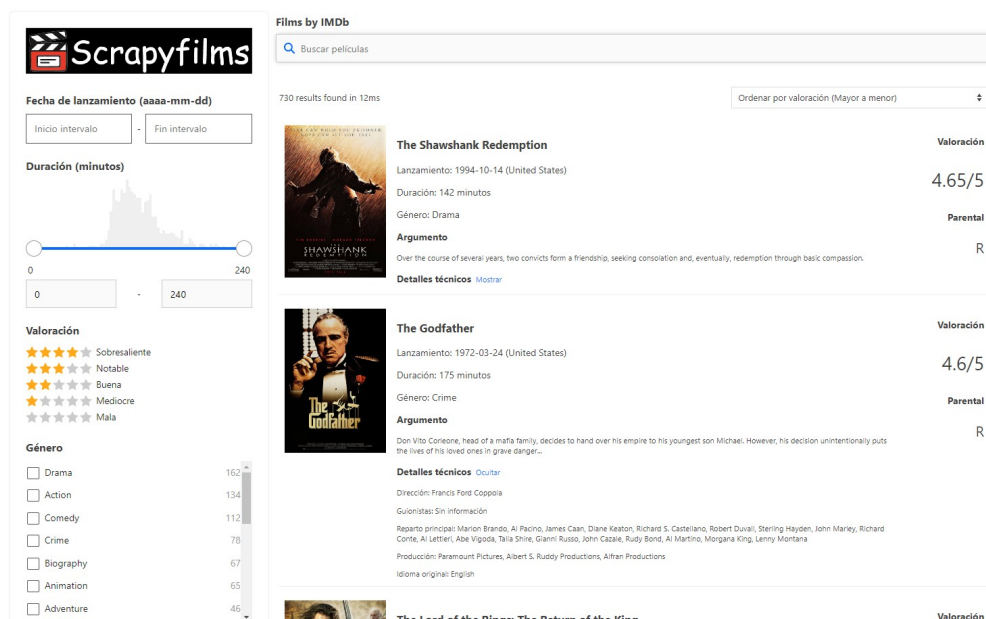


Figura 5: Vista inicial de la web.

Para una correcta conexión entre la web y el motor de búsqueda se tuvo que editar la configuración de *ElasticSearch* editando el archivo *elasticsearch.yml* (el cual también está disponible para su consulta en el repositorio del proyecto) de forma que se ha eliminado la seguridad y se ha permitido el CORS y las conexiones al puerto de la web (en nuestro caso el 3000):

```
# Enable security features
xpack.security.enabled: false
```

Figura 6: Configuración de seguridad para ElasticSearch.

```
http.cors.enabled: true
http.cors.allow-origin: "http://localhost:3000"
```

Figura 7: Configuración de CORS para ElasticSearch.

## Filtros utilizados

**Fecha de lanzamiento (aaaa-mm-dd)**

Inicio intervalo - Fin intervalo

**Duración (minutos)**

0 240

0 240

**Valoración**

★★★★★ Sobresaliente  
★★★★☆ Notable  
★★★☆☆ Buena  
★★☆☆☆ Mediocre  
★☆☆☆☆ Mala

**Género**

<input type="checkbox"/> Drama	162
<input type="checkbox"/> Action	134
<input type="checkbox"/> Comedy	112
<input type="checkbox"/> Crime	78
<input type="checkbox"/> Biography	67
<input type="checkbox"/> Animation	65
<input type="checkbox"/> Adventure	46

**País de origen**

<input type="checkbox"/> United States	528
<input type="checkbox"/> United Kingdom	68
<input type="checkbox"/> France	20
<input type="checkbox"/> Japan	17
<input type="checkbox"/> Germany	11
<input type="checkbox"/> India	11
<input type="checkbox"/> Italy	11

**Idioma original**

<input type="checkbox"/> English	638
<input type="checkbox"/> Japanese	15
<input type="checkbox"/> French	13
<input type="checkbox"/> Spanish	9
<input type="checkbox"/> Hindi	8
<input type="checkbox"/> German	7
<input type="checkbox"/> Italian	7

**Guía parental**

<input type="checkbox"/> R	287
<input type="checkbox"/> PG-13	158
<input type="checkbox"/> PG	121
<input type="checkbox"/> Not Rated	36
<input type="checkbox"/> Approved	30
<input type="checkbox"/> Passed	29
<input type="checkbox"/> G	26

Figura 8: Filtros de búsqueda de la web.

**Rango de fechas (DateRange):** Permite seleccionar películas publicadas entre un intervalo de fechas.

**Duración en minutos (RangeInput):** Se selecciona un intervalo de duración de las películas en minutos.

**Valoración (RatingsFilter):** Se pueden filtrar las películas por valoración (del 0 al 5 en intervalos de 1 en 1). No permite varias valoraciones simultáneas.

**Lista de géneros (MultiList):** Muestra una lista de los diferentes géneros disponibles permitiendo al usuarios la selección múltiple.

**Lista de países de origen (MultiList):** Muestra una lista de los diferentes países de origen disponibles permitiendo al usuarios la selección múltiple.

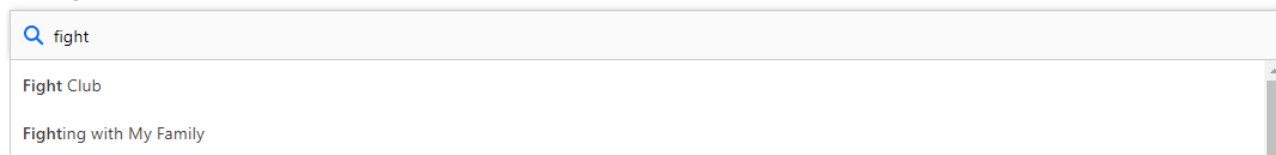
**Lista de idiomas originales (MultiList):** Muestra una lista de los diferentes idiomas originales disponibles permitiendo al usuarios la selección múltiple.

**Lista de guías parentales (MultiList):** Muestra una lista de las diferentes guías parentales disponibles permitiendo al usuarios la selección múltiple.

## Barra de búsqueda y lista de filtros activos

La web también dispone de una barra de búsqueda en la que podremos buscar películas por su título, pero también en función de actores o directores. La barra de búsqueda está configurada para que haga sugerencias en función del contenido escrito en ella:

### Films by IMDb

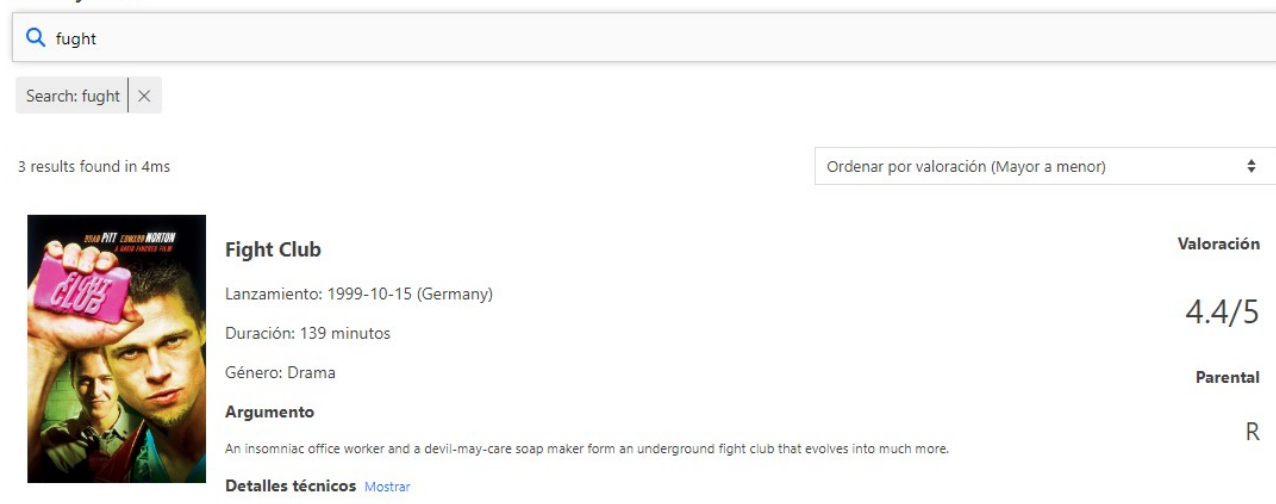


A search bar with a magnifying glass icon and the text 'fight'. Below the bar, a list of suggestions is visible: 'Fight Club' and 'Fighting with My Family'.

Figura 9: Autocompletado de la barra de búsqueda.

La barra de búsqueda también permite obtener resultados con hasta 1 carácter diferente (para casos de errores tipográficos como “fight” y “fught”):

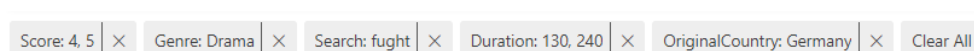
### Films by IMDb



A search bar with a magnifying glass icon and the text 'fught'. Below the bar, a button says 'Search: fught' with a close icon. Below that, it says '3 results found in 4ms'. To the right, a dropdown menu shows 'Ordenar por valoración (Mayor a menor)'. Below this, a movie card for 'Fight Club' is displayed. The card includes a poster, the title 'Fight Club', release date 'Lanzamiento: 1999-10-15 (Germany)', duration 'Duración: 139 minutos', genre 'Género: Drama', and a synopsis: 'An insomniac office worker and a devil-may-care soap maker form an underground fight club that evolves into much more.' The rating is '4.4/5' and the parental rating is 'R'. At the bottom of the card, it says 'Detalles técnicos' and 'Mostrar'.

Figura 10: Resultados de una búsqueda con un error tipográfico.

Finalmente, bajo la barra de búsqueda se muestra un listado de los filtros que están activos en la búsqueda actual con posibilidad de eliminarlos:



A horizontal bar containing several filters: 'Score: 4, 5', 'Genre: Drama', 'Search: fught', 'Duration: 130, 240', 'OriginalCountry: Germany', and a 'Clear All' button. Each filter has a close icon.


Figura 11: Lista de filtros activos.

## Lista de resultados

Una vez realizada una búsqueda, los resultados se listan bajo la barra de búsqueda y la lista de filtros. Los resultados son paginados en grupos de 10 elementos, y cada uno muestra el póster de la película junto a su información. Para una interfaz menos cargada, la información más técnica de cada película aparece oculta y se puede mostrar mediante un botón:

1 results found in 3ms

Ordenar por valoración (Mayor a menor)



### Shutter Island

Lanzamiento: 2010-02-19 (United States)

Duración: 138 minutos

Género: Mystery

**Argumento**

Teddy Daniels and Chuck Aule, two US marshals, are sent to an asylum on a remote island in order to investigate the disappearance of a patient, where Teddy uncovers a shocking truth about th...

**Detalles técnicos** Ocultar

Dirección: Martin Scorsese

Guionistas: Laeta Kalogridis, Dennis Lehane

Reparto principal: Leonardo DiCaprio, Emily Mortimer, Mark Ruffalo, Ben Kingsley, Max von Sydow, Michelle Williams, Patricia Clarkson, Jackie Earle Haley, Ted Levine, John Carroll Lynch, Elias Koteas, Robin Bartlett, Christopher Denham, Nellie Sciutto, Joseph Sikora, Curtiss Cook, Raymond Anthony Thomas, Joseph McKenna

Producción: Paramount Pictures, Phoenix Pictures, Sikelia Productions

Idioma original: English

Valoración

4.1/5

Parental

R

Prev 1 Next

Figura 12: Información de una película con sus detalles técnicos.

La lista de resultados indica en la parte superior cuántas películas entran en los filtros marcados y estas se pueden ordenar de diversas formas, siendo el método de ordenación por defecto “Por relevancia”, que ordena los resultados en función del campo “\_score”, que indica la relevancia de ese objeto con la búsqueda actual.

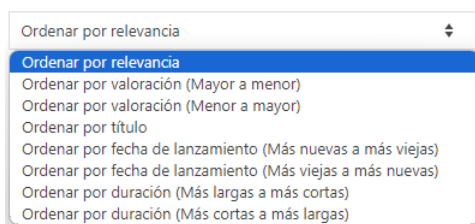


Figura 13: Lista de métodos de ordenación.

En el objeto *ReactiveList*, encargado de listar los resultados, se transfiere el *renderizado* de cada ítem a la función *RenderFilm(res)* que obtiene los datos de cada película, los procesa para evitar errores y genera cada ítem de la lista de forma correcta.

# Ejecución y acceso al proyecto

## Ficheros y carpetas relevantes

En el repositorio de GitHub del proyecto hay una serie de ficheros y carpetas necesarios para una correcta ejecución de la práctica:

- ***dumps/*** : Carpeta que contiene los ficheros *films.json* y *films\_formatted.json*. En el caso del volcado de datos a *ElasticSearch* será necesario usar el fichero formateado. El fichero *films.json* solamente es una muestra del resultado del *crawling* sin su correspondiente filtrado y preparado.
- ***scrapyfilms.postman\_collection.json***: Colección de las peticiones *HTTP* para *Postman* utilizadas en el proyecto.
- ***elasticsearch.yml***: Archivo de configuración de *ElasticSearch* para el correcto funcionamiento con la web.

## Ejecución del proyecto

Para la ejecución, se supone que se el equipo a emplear tiene *ElasticSearch* instalado y configurado como se ha explicado anteriormente y en ejecución en el puerto 9200.

1. Clonado del proyecto: ***git clone https://github.com/joaquinsolla/scrapyfilms***
2. Nos colocamos en la carpeta raíz del proyecto: ***cd scrapyfilms***
3. Ejecución del *spider*: ***scrapy crawl no-scraper -o dumps/films.json***
4. Nos colocamos en la carpeta del *crawler*: ***cd scrapyfilms***
5. Ejecutamos el formateador para *Postman*: ***python postman\_formatter.py***
6. Ejecución de las peticiones 01 a 07 (incluidas) en *Postman*.
7. Nos colocamos en la carpeta del proyecto web: ***cd ../scrapyfilms-web***
8. Ejecutamos la aplicación web: ***npm start***

Seguidos estos pasos, la aplicación será accesible en cualquier navegador en la dirección <http://localhost:3000/>.

\* *La ejecución del spider puede ser muy larga (de más de una hora), por lo que se proporcionan ya los archivos *films.json* y *films\_formatted.json* en la carpeta *dumps* del proyecto. De esta forma podemos pasar directamente del paso 2 al 6.*

## Acceso al proyecto

Todo el código del proyecto está público y disponible en *GitHub*, en el enlace <https://github.com/joaquinsolla/scrapyfilms>.

\* *En los contribuidores del repositorio aparece un integrante que no ya figura en el grupo de trabajo (moliveirac) pero que antes sí, ya que se dio de baja del máster al comienzo del proyecto.*

## Conclusiones y trabajo futuro

Con el desarrollo de este proyecto se han obtenido nuevos conocimientos en el campo de los motores de búsqueda y las consultas sobre ellos, además del uso de *ElasticSearch* y de la librería *ReactiveSearch*.

El proyecto podría contar con actualizaciones futuras y mejoras, estas son algunas de las planteadas:

- Inclusión de nuevas películas.
- Accesibilidad de la web (idiomas, formatos de fecha...).
- Una interfaz más elaborada.
- Nuevos campos de información para las películas.