

TRABAJO PRÁCTICO INTEGRADOR

IA 3.5 - Redes de Datos

Tecnicatura Universitaria en Inteligencia Artificial

Accurso, Agustín - Mancini, Nicolás

26/06/2025

Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

INTRODUCCIÓN

Para el siguiente trabajo se desarrolló una API REST utilizando el framework FastAPI en Python, como base de datos se usó una de películas en formato JSON disponible públicamente en GitHub. El sistema fue implementado sobre Windows 10, ejecutado en un entorno virtual creado con Anaconda, y resumidamente, la API permite realizar búsquedas, consultas y modificaciones sobre los datos de las películas.

DESCRIPCIÓN DEL ENTORNO DE TRABAJO DEL SERVIDOR

El entorno del servidor fue configurado con Anaconda Navigator, donde se creó un entorno virtual con Python 3.12.11. Se instalaron los paquetes necesarios utilizando pip, incluyendo fastapi, uvicorn, pydantic y requests, y se utilizó el editor Visual Studio Code para escribir y ejecutar el código.

Durante la configuración no se presentaron grandes dificultades, pero fue importante seleccionar correctamente el intérprete de Python dentro de VSCode para que coincidiera con el entorno virtual creado.

DESCRIPCIÓN DEL ENTORNO DE TRABAJO DEL CLIENTE

El cliente accede a la API desde el navegador web a través de la interfaz generada automáticamente por FastAPI (/docs) o desde herramientas como Postman. No se requiere ninguna instalación adicional del lado del cliente. Las pruebas se realizaron en el mismo equipo local, por lo que no hubo problemas de red o autenticación complejos.

Se implementó autenticación básica (HTTPBasic) para proteger las operaciones de escritura (POST, PUT, DELETE).

DESCRIPCIÓN DE LA BASE DE DATOS ELEGIDA

Se utilizó el archivo JSON disponible en el siguiente enlace: <https://raw.githubusercontent.com/prust/wikipedia-movie-data/master/movies.json>

Este archivo contiene aproximadamente 36.273 objetos, cada uno representando una película. Cada película tiene las siguientes propiedades:

- **title** (str): el nombre de la película.
- **year** (int): año de lanzamiento.
- **cast** (List[str]): lista de actores.
- **genres** (List[str]): lista de géneros (comedia, acción, etc).
- **href** (str, opcional): identificador de la película en Wikipedia, no es la URL directa. Es un texto que puede ser usado para construir enlaces como <https://en.wikipedia.org/wiki/{href}>
- **extract** (str, opcional): descripción corta de la película.
- **thumbnail** (str): URL a una imagen (miniatura) de la película o relacionada.
- **thumbnail_width** (int, opcional): ancho en píxeles de la imagen.

- **thumbnail_height** (int, opcional): alto en píxeles de la imagen.

PROCEDIMIENTO

Esta API tiene como finalidad ofrecer una interfaz para consultar, gestionar y modificar información sobre películas. Además, permite a los usuarios mantener una watchlist personalizada, en la que pueden agregar, ver y eliminar películas que desean ver más adelante. La API está pensada para ser simple, rápida y funcionar completamente en memoria, es decir, almacena los cambios de modo temporal.

Posibles consultas, con ejemplos:

Buscar título: permite obtener toda la información disponible de una película específica a partir de su título.

```
{  
  "title": "Once Upon a Time in Hollywood",  
  "year": 2019,  
  "cast": [  
    "Leonardo DiCaprio",  
    "Brad Pitt",  
    "Margot Robbie",  
    "Al Pacino",  
  ],  
  "genres": [  
    "Comedy",  
    "Drama"  
  ],  
  "href": "Once_Upon_a_Time_in_Hollywood",  
  "extract": "Once Upon a Time...in Hollywood is a 2019 comedy-drama film written and directed by Quentin Tarantino. Produced by Columbia Pictures, Bona Film Group, Heyday Films, and Visiona Romantica and distributed by Sony Pictures Releasing, it is a co-production between the United States, United Kingdom, and China. It features a large ensemble cast led by Leonardo DiCaprio, Brad Pitt, and Margot Robbie. Set in 1969 Los Angeles, the film follows a fading actor and his stunt double as they navigate the rapidly changing film industry, with the looming threat of the Tate murders hanging overhead. It features \"multiple storylines in a modern fairy tale tribute to the final moments of Hollywood's golden age.\""  
  "thumbnail":  
    "https://upload.wikimedia.org/wikipedia/en/a/a6/Once_Upon_a_Time_in_Hollywood_poster.png",
```

"thumbnail_width": 259,

"thumbnail_height": 384

}

Buscar actor: permite ingresar el nombre de un actor o actriz y obtener una lista de todas las películas en las que participa dentro del catálogo.

['Cutting Class', 'Across the Tracks', 'Johnny Suede', 'Thelma & Louise', 'Cool World', 'A River Runs Through It', 'Kalifornia', 'True Romance', 'The Favor', 'Interview with the Vampire', 'Legends of the Fall', '12 Monkeys', 'Seven', 'Sleepers', 'The Devil's Own', 'Seven Years in Tibet', 'Meet Joe Black', 'Fight Club', 'Snatch', 'The Mexican', 'Ocean's Eleven', 'Spy Game', 'Sinbad: Legend of the Seven Seas', 'Troy', 'Ocean's Twelve', 'Mr. & Mrs. Smith', 'Babel', 'Ocean's Thirteen', 'The Assassination of Jesse James by the Coward Robert Ford', 'Burn After Reading', 'The Curious Case of Benjamin Button', 'Inglourious Basterds', 'Megamind', 'The Tree of Life', 'Moneyball', 'Happy Feet Two', 'Killing Them Softly', 'World War Z', '12 Years a Slave', 'The Counselor', 'Fury', 'By The Sea', 'The Big Short', 'Allied', 'War Machine', 'Once Upon a Time in Hollywood', 'Ad Astra', 'The Lost City', 'Bullet Train', 'Babylon']

Total: devuelve la cantidad de películas que hay actualmente cargadas en el memoria, para conocer el tamaño del catálogo.

36273

Obtener cantidad según género: permite consultar cuántas películas de determinado género específico (por ejemplo, "Acción", "Drama", "Comedia") tenemos en el catálogo.

"Hay 14062 peliculas de Drama"

Agregar película: permite incorporar una nueva película al catálogo a partir de un parámetro que recibe un archivo en formato JSON con todos los datos requeridos. Esta operación requiere autenticación del usuario.

Modificar película: permite cambiar un atributo específico de una película ya existente, como su año, elenco o género. Se deben indicar el nombre de la película, el atributo a modificar y el nuevo valor. Esta operación también requiere autenticación.

Eliminar película: elimina una película del catálogo a partir de su título. También requiere autenticación del usuario.

Agregar watchlist: agrega una película específica a la lista personal del usuario. Se debe indicar el título de la película, que debe existir previamente en el sistema.

"Película 'Once upon a time in hollywood' agregada a la watchlist."

Obtener watchlist: devuelve la lista de películas que el usuario ha guardado en su watchlist personal.

["Once Upon a Time in Hollywood","The Godfather"]

Eliminar watchlist: permite quitar una película de la watchlist del usuario, indicando su nombre siempre y cuando la película esté en la lista.

INSTRUCCIONES DE USO DEL CLIENTE

El archivo cliente.py actúa como cliente de una API que permite gestionar un catálogo de películas y una watchlist. Al ejecutarlo, el usuario puede interactuar con la API de forma sencilla mediante un menú de opciones en la terminal. Al iniciar, el usuario debe ingresar su nombre de usuario y contraseña, para determinar si puede acceder a las funcionalidades que requieren autenticación, como agregar, modificar o eliminar películas del catálogo. Dependiendo de la opción seleccionada en el menú, el programa solicitará uno o más parámetros al usuario como el nombre de una película, entre otros datos que son necesarios para que la API procese correctamente la solicitud al pasarlos como parámetros.