

# Acerca de la resolución de un problema: Linea de comando vs Python

Kevin García Mundo

30 de septiembre de 2024

A continuación haremos una comparativa sobre la resolución de un problema dado usando linea de comando y python.

El problema es el siguiente:

“Utilizando la línea de comandos y los comandos ya incluidos en la distribución de Linux que instalaste en tu máquina virtual, encuentra las 10 películas peor calificadas en promedio y las 10 películas mejor calificadas en promedio del directorio train. No es necesario encontrar estas respuestas con un solo comando, puedes utilizar varios para lograr el cometido. Puede resultar difícil obtener programáticamente los nombres de las películas, es suficiente poder identificarlas utilizando las direcciones URL presentes en los archivos urls\_neg.txt y urls\_pos.txt.

Sugerencia: Investiga sobre los comandos grep, sed y awk, son más complicados que los comandos discutidos en clases.”

Posteriormente se nos asigno la tarea de resolver el mismo problema utilizando python.

Aquí mi discusión, desde mi perspectiva, para ambas soluciones presentadas.

## 1. Linea de comando

Si bien la linea de comando en un primer contacto resulta realmente desafiante por no tener una interfaz tan amigable y no tener la posibilidad de visualizar lo que estas haciendo con cada comando que ejecutas, para llegar a la solución basta con ejecutar cinco lineas con las herramientas adecuadas.

Al realizar esta actividad se necesitó de investigar comandos que te permitieran hacer lo que tenias en mente, pero tenerlo en mente no quiere decir que tal comando exista, por lo que debías hacerlo secuencialmente o pensar en alguna otra manera de atacar el problema.

Ventajas:

- Teniendo el conocimiento sobre que comandos usar el problema se puede resolver en pocos minutos.

- Solamente es necesario disponer de la terminal.
- Los comandos se ejecutan con rapidez.

Desventajas:

- No cuentas con una interfaz amigable para visualizar lo que estás haciendo.
- Los comandos suelen ser difíciles de recordar.

```

kevin@kevin-VirtualBox: ~/Escritorio/ClaseProg/A-02/aclmdb/train
kevin@kevin-VirtualBox: ~/Escritorio/ClaseProg/A-02/aclmdb/train$ awk '{ puntos[$1] += $2; suma[$1]++ } END { for (url in puntos)
print url, puntos[url]/suma[url] }' url_cal_pos.txt | sort -k2,2nr | head -n 10 > promedio_pos.txt
kevin@kevin-VirtualBox: ~/Escritorio/ClaseProg/A-02/aclmdb/train$ cat promedio_pos.txt
http://www.imdb.com/title/tt0024675/usercomments 10
http://www.imdb.com/title/tt0027791/usercomments 10
http://www.imdb.com/title/tt0031335/usercomments 10
http://www.imdb.com/title/tt0033275/usercomments 10
http://www.imdb.com/title/tt0052855/usercomments 10
http://www.imdb.com/title/tt0052992/usercomments 10
http://www.imdb.com/title/tt0061904/usercomments 10
http://www.imdb.com/title/tt0064750/usercomments 10
http://www.imdb.com/title/tt0067786/usercomments 10
http://www.imdb.com/title/tt0068235/usercomments 10
kevin@kevin-VirtualBox: ~/Escritorio/ClaseProg/A-02/aclmdb/train$

```

Figura 1: Ejemplo de solución con línea de comando.

## 2. Python

En python resultó ser muy distinto a lo comentado en la sección anterior, personalmente las instrucciones implementadas en el código no eran del todo desconocidas pero el código creció de una manera que no esperaba.

Primero se elaboró una solución sin usar programación orientada a objetos, una vez terminada no sería tan difícil pasar a un módulo todo lo que se hizo pues cada paso lo hice usando funciones distintas.

En este caso, usando Docker puedes ver todo lo que ocurre con las carpetas que se crean y, usando pandas, puedes visualizar si lo que estás haciendo lleva el curso deseado, además hay varias formas de presentar o imprimir la solución que puedes controlar de manera sencilla.

Ventajas:

- Puedes visualizar fácilmente las salidas de tus códigos.
- Toda la documentación de las librerías y/o módulos que poseeé python.
- El manejo de excepciones y una explicación de los errores.

Desventajas:

- Necesitas instalar varias librerías que python no tiene por defecto.
- El código creció (demasiado) en comparación de la solución propuesta con linea de comando.
- El código tarda en finalizar la tarea.

```

top-impdb-py3.12 PS C:\Users\DELL\OneDrive\top-impdb\programa\top-impdb> python -m python_scripts.ejecutar_analisis
El archivo ya existe.
Descargando en: /Downloads/basesdatos/ac1Impdb.
Las 10 películas mejor calificadas:
Película: Atlantis: El imperio perdido (2001) | Promedio: 10.00 | url: http://www.imdb.com/title/tt0238011
Película: Garden of Allah (1936) | Promedio: 10.00 | url: http://www.imdb.com/title/tt0192745
Película: El Jardín de la Alegría (2000) | Promedio: 10.00 | url: http://www.imdb.com/title/tt0195234
Película: The Thing About My Folks (2005) | Promedio: 10.00 | url: http://www.imdb.com/title/tt0429177
Película: The Cune (Short 2006) | Promedio: 10.00 | url: http://www.imdb.com/title/tt0956331
Película: Dark Angel: Elites (TV Series 2000) | Promedio: 10.00 | url: http://www.imdb.com/title/tt01554587
Película: Ned Kelly (2003) | Promedio: 10.00 | url: http://www.imdb.com/title/tt0277941
Película: Hechizo de Luna (1987) | Promedio: 10.00 | url: http://www.imdb.com/title/tt0093565
Película: The Last Days of Pompeii (TV Series 1999) | Promedio: 10.00 | url: http://www.imdb.com/title/tt01719029
Película: Bourne: El ultimátum (2007) | Promedio: 10.00 | url: http://www.imdb.com/title/tt0849693
Las 10 películas peor calificadas:
Película: Reunión (TV Series 2005-2006) | Promedio: 1.00 | url: http://www.imdb.com/title/tt0460672
Película: El gato (2003) | Promedio: 1.00 | url: http://www.imdb.com/title/tt0312528
Película: Titanic (1997) | Promedio: 1.00 | url: http://www.imdb.com/title/tt0120338
Película: Do Not Fall in Love (2006) | Promedio: 1.00 | url: http://www.imdb.com/title/tt02012334
Película: Mi novia es canibal (1988) | Promedio: 1.00 | url: http://www.imdb.com/title/tt0095556
Película: Tashan (2008) | Promedio: 1.00 | url: http://www.imdb.com/title/tt0995572
Película: Oil Warning (1999) | Promedio: 1.00 | url: http://www.imdb.com/title/tt0183976
Película: La casa de los espíritus (1993) | Promedio: 1.00 | url: http://www.imdb.com/title/tt0107151
Película: El culto siniestro (2006) | Promedio: 1.00 | url: http://www.imdb.com/title/tt0450345
Película: Annie: A Royal Adventure (TV Movie 1995) | Promedio: 1.00 | url: http://www.imdb.com/title/tt0112374

```

Figura 2: Ejemplo de solución con python.

### 3. Conclusión

La línea de comandos ofrece una solución rápida y eficiente para tareas específicas, como encontrar las películas mejor y peor calificadas, utilizando herramientas como `grep`, `sed` o `awk`. Su principal ventaja es la velocidad de ejecución y la simplicidad en la cantidad de comandos necesarios. Sin embargo, carece de una interfaz amigable para visualizar resultados y tiene una curva de aprendizaje más pronunciada, lo que puede dificultar su uso si no se está familiarizado con los comandos.

Por otro lado, Python ofrece una mayor flexibilidad y capacidad para personalizar la solución, permitiendo el uso de bibliotecas como pandas para analizar y visualizar los datos. Aunque el código tiende a ser más largo y su ejecución puede ser más lenta en comparación con la línea de comandos, brinda más control sobre el flujo del proceso y facilita el manejo de excepciones y errores. Python es ideal para proyectos que requieren un análisis más detallado, pero a costa de una mayor complejidad en su configuración y uso.