

Programación Técnica y Científica.

Práctica II. Gestión de información.

1.- Descripción del problema.

Existe una base de datos sobre películas conocida como IMDB (Internet Movie DataBase). Esta base de datos está disponible en formato txt y por tanto es fácil de descargar y procesar usando python. En formato txt se pueden encontrar en www.imdb.com/interfaces#plain y aparecen distintos ficheros relativos a los diferentes aspectos recogidos en la base de datos. Para esta práctica vamos a utilizar los ficheros actors.list.gz y actresses.list.gz que corresponden a la información recopilada sobre actores y actrices.

Implementa un script de python capaz de descargar automáticamente ambos ficheros y procesar la información teniendo en cuenta el formato. Una vez procesada se debe poder listar para cada actor/actriz el título de cada una de las películas en las que aparece, la fecha y el papel que tiene. Debe considerar la posibilidad de comprobar periódicamente si esos ficheros se han actualizado.

También debes crear una base de datos con toda esa información usando shelve y que permita dado un actor/actriz conocer todas su películas.

Escribe una función que dados dos nombres de actores/actrices devuelva los datos de las películas en las que han sido coprotagonistas.

El actor Kevin Bacon afirmó en una ocasión que había trabajado con todo el que era alguien importante en el mundo del cine de forma directa o siguiendo esa relación “trabajar con” con un número mínimo de relaciones. A este problema se le conoce como número de Bacon (http://es.wikipedia.org/wiki/Número_de_Bacon) y está relacionado con el problema de los seis grados de separación (http://es.wikipedia.org/wiki/Seis_grados_de_separación). Escribe una función que determine el número de grados de separación entre dos actores/actrices y para ello usa un algoritmo de camino más corto (http://es.wikipedia.org/wiki/Problema_del_camino_más_corto).

2.- Entrega.

Se deben entregar los siguientes archivos:

- descargar_imdb.py
- peliculas_actor.py
- coprotagonistas.py
- grados_separacion.py
- utilidades.py

Las soluciones deben ser eficientes.