P36 – Reto 5 Programando para NetTicx

La empresa de video en streaming **NetTicx** ha incrementado significativamente la cantidad de películas en su catálogo durante los últimos meses, por lo que ha tenido problemas para gestionar su información.

La empresa, en sus bases de datos internas representa cada película con un número entero y con esta información es usual que se realicen tareas como, eliminar duplicados sobre un listado de películas, encontrar películas que pertenecen a ciertas categorías, permitir a sus usuarios intercambiar películas, entre otras.

Hasta el momento, todas estas tareas se realizan de forma manual por lo que la empresa quiere empezar a usar los beneficios de la programación para que todas estas se puedan realizar de forma automática.

Usted es contratado para realizar un módulo/librería de Python que tenga las funciones requeridas para ejecutar automáticamente las tareas de procesamiento de información realizadas por la empresa.

Se describen a continuación, cada una de las funciones que deben estar incluidas el módulo/librería que usted va a desarrollar.

Nombre del módulo/librería: utilsvideo

Funciones que debe contener la librería “utilsvideo”

1. Nombre de la función: peliculas_sin_repeticion
   Descripción: la función debe recibir una lista de películas E1 (representadas por números enteros) y debe eliminar las que se encuentren repetidas, retornando una lista donde cada película aparece una sola vez.
   Entradas:
   - E1: (lista) Lista de películas

   Retorno:
   - (lista) Lista de películas sin repetición

   Ejemplo:
   El llamado a la función peliculas_sin_repeticion ([8, 7, 8, 2, 3, 2, 1, 3])
   debería retornar [8, 7, 2, 3, 1]

2. Nombre de la función: pelicula_en_posiciones
   Descripción: A partir de una película E3, una lista de películas E2 y una lista de posiciones E1, se desea conocer, de la lista de posiciones E1, cuáles de estas

posiciones corresponden a la película E3 si buscamos en la lista de películas E2 en dicha posición. La función debe retornar una lista con los elementos de E1 que al acceder a ellos (a su posición) en E2 correspondan a la película E3. Ver ejemplo.

Entradas:

- E1: (lista) Lista de posiciones
- E2: (lista) Lista de películas
- E3: (entero) Película

Retorno:

- (lista) Lista de posiciones seleccionadas de acuerdo con la descripción

Ejemplo:

El llamado a la función pelicula_en_posiciones ([1, 2, 5], [4, 4, 6, 2, 1, 4], 4) debería retornar [1, 5], porque la película **4** se encuentra en la posición 1 y también en la posición 5 pero NO en la posición 2 de la lista de películas.

El llamado a la función pelicula_en_posiciones([0, 2, 3, 6], [7, 8, 9, 7, 9, 9, 9, 8], 9) debería retornar [2, 6], porque la película **9** se encuentra en la posición 2 y también en la posición 6 pero NO en la posición 0 ni en la 3 de la lista de películas.

3. Nombre de la función: solo_drama

   Descripción: A partir de la lista de películas de drama (E1) y la lista de películas de amor (E2) se desea conocer las películas que pertenecen únicamente a la categoría drama. La función debe retornar una lista con las películas que pertenecen **únicamente** a la categoría drama.

   Entradas:

   - E1: (lista) Lista de películas de categoría drama
   - E2: (lista) Lista de películas de categoría amor

   Retorno:

   - (lista) Lista de películas que pertenecen únicamente a la categoría drama

   Ejemplo:

   El llamado a la función solo_drama ([5, 1, 4], [2, 1])
   debería retornar [5, 4]
   El llamado a la función solo_drama ([3,5,7,10,15,16], [4,10,5,8])
   debería retornar [3,7,15,16]

4. Nombre de la función: numero_cambios

   Descripción: La empresa también permite a sus usuarios intercambiar películas que ya fueron vistas. Por lo tanto, se requiere que a partir de las listas de películas que ya fueron vistas por dos usuarios (lista E1 y lista E2) se calcule el número de películas que pueden intercambiar estos dos usuarios entre ellos. La función debe recibir las listas de películas que ya fueron vistas por dos usuarios y debe retornan la cantidad de películas que pueden intercambiar, teniendo en cuenta que el cambio se hace 1 a 1 (se cambia 1 película por otra) y que los dos están interesados en cualquier película menos en las que están en su propia lista.

   Entradas:

   - E1: (lista) Lista de películas que el usuario1 ya vio

- E2: (lista) Lista de películas que le usuario2 ya vio

Retorno:

- (entero) cantidad de películas que pueden intercambiar

Ejemplo:

El llamado a la función numero_cambios([3, 5, 7, 10, 15, 16], [4, 10, 5, 8]) debería retornar 2, porque al primer usuario solo le interesan dos películas del segundo usuario: [4,8] (las que él no tiene pero el otro sí), mientras que al segundo usuario solo le interesan las películas: [3,7,15,16]. Por lo tanto, cómo al primer usuario le interesan 2 películas y al segundo le interesan 4, únicamente se pueden realizar 2 intercambios.

Realice un módulo/librería en Python que contenga las 4 funciones descritas previamente.

IMPORTANTE: el nombre del módulo/librería y de las funciones debe ser igual al que está descrito en las especificaciones previas.

Calificación:

El juez comprobará la existencia del módulo/librería y ejecutará varios casos de prueba por cada una de las 4 funciones definidas. La calificación será sobre 5 puntos que se obtienen de la siguiente forma:

1 punto: Existencia del módulo/librería “utilsvideo” (debe crear un archivo con el nombre utilsvideo.py que contenga las funciones)
1 punto: Definición apropiada de la función “peliculas_sin_repeticion” dentro del módulo
1 punto: Definición apropiada de la función “pelicula_en_posiciones” dentro del módulo
1 punto: Definición apropiada de la función “solo_drama” dentro del módulo
1 punto: Definición apropiada de la función “numero_cambios” dentro del módulo