



Algoritmos y Programación II

Trabajo práctico N° 3:

“Listas con listas”

Objetivo

Desarrollar un programa orientado a objetos en C++ que maneje listas que a su vez contienen listas.

Datos de entrada

Los datos de entrada serán dos archivos de texto *películas_vistas.txt* y *películas_no_vistas.txt*.

Ambos archivos tienen el mismo formato:

Nombre_pelicula

Género

Puntaje

Director/a

actor/actriz_1 *actor/actriz_2...* *actor/actriz_n*

Ejemplo:

Mi obra maestra

Comedia negra

8

Gaston Duprat

Luis_Brandoni Lucas_Aranda Raúl_Arévalo Mahmoud_Azim Daniela_Katz Guillermo_Francella Andrea_Frigerio

I, como Icaro

Drama Suspenso

10

Henri Verneuil

Yves_Montand Michel_Albertini Roland_Amstutz Jean-Pierre_Bagot

Jack and Jill

Comedia

3

Dennis Dugan

Adam_Sandler Al_Pacino Katie_Holmes

Carmen y Lola

Drama

7

Arantxa Echevarria

Zaira_Romero Rosy_Rodriguez Rafaela_Leon Carolina_Yuste

Cementerio de animales

Terror

7

Kevin Kolsch

Jason_Clarke Amy_Seimetz John_Lithgow

Requisitos

La aplicación debe leer estos archivos y generar dos listas: películas_vistas y películas_no_vistas. Luego ofrecerá un menú para mostrarlas por separado. Además debe elaborar y mostrar una tercera lista: películas_recomendadas. Esta lista se armará de la siguiente forma:

- Con cada película vista se tomará: el género, el director, y los actores.
- Comparando en las no vistas, si **hay coincidencia en el género y [(en el director o por lo menos uno de los actores) o el puntaje es 7 o más]**, se toma la película como recomendada.

Consideraciones

- La aplicación debe estar o completamente orientada a objetos o ser un híbrido, admitiendo solo el menú y algunas funciones menores como desarrollo estructurado.
- Los archivos están bien formados.
- Las listas son dinámicas
- Las listas utilizan templates
- La cantidad de actores que intervienen en una película es variable
- No se pueden utilizar objetos de STL.

Qué se evalúa

- Buenas prácticas: código, modularización, comentarios, claridad
- POO
- Funcionalidad
- Uso de templates
- Memoria dinámica
- Interfaz de usuario
- Pre y pos condiciones

Normas de entrega

El ejercicio debe ser desarrollado en equipos de cuatro o cinco personas.

Deben entregar:

- Documentación
 - o Diagrama de clases UML
 - o Diagrama de relación de clases.
 - o Descripción de cada TDA, indicando las pre y poscondiciones de cada una de las operaciones.

- Código fuente
 - En los archivos .h también van las pre y poscondiciones.
- Archivos para testeo.

La fecha de entrega vence el lunes 3/6 a las 23.55hs