

Tema 2 – Recuperatorio Práctico Unidad 2

Resuelva la siguiente situación problemática, utilizando el lenguaje de programación Python.

La plataforma “netView” permite a sus miembros ver películas en cualquier momento. Usted ha sido contratado para desarrollar un sistema que permita administrar la información de las visualizaciones de las películas que realizan los miembros de la plataforma.

Para ello cuenta con los datos de:

Cada Miembro: identificador del miembro, apellido y nombre, correo electrónico.

Cada Visualización: identificador del miembro, identificador de la película que vio, fecha y hora de visualización, minutos que duró la visualización.

Los datos de los miembros están registrados en el archivo “Miembros.csv”; y los datos de las visualizaciones en el archivo “Visualizaciones.csv”. Ambos archivos tienen como separador el carácter “;”.

El equipo de desarrollo ha acordado que cada clase debe estar en un módulo, y usted debe implementar en Python:

1. Las clases Miembro y Visualizacion, con los atributos y métodos necesarios.
2. Una clase GestorMiembros que almacene instancias de la clase Miembro. Cada una de estas instancias se creará con los datos registrados en el archivo “Miembros.csv”. Este gestor debe basarse en un arreglo Numpy.
3. Una clase GestorVisualizaciones que almacene instancias de la clase Visualizacion. Cada una de estas instancias se creará con los datos registrados en el archivo “Visualizaciones.csv”. Este gestor debe basarse en una lista Python.
4. La carga de cada gestor con los datos respectivos.
5. Utilizando exclusivamente los Gestores cargados con los datos de los archivos, implemente un menú de opciones que permita:
 - a. Ingresar el correo electrónico de un miembro e indicar la cantidad total de minutos que ha visto películas.
 - b. Mostrar apellido, nombre y correo electrónico de los miembros que han realizado visualizaciones simultáneas. Es decir, para el mismo miembro, en la misma fecha y a la misma hora hay registradas más de una visualización.

Reglas:

- Para esta funcionalidad debe definir la sobrecarga del operador `__eq__` en la clase Visualizacion considerando el identificador del miembro, la fecha y hora de la visualización.
- Resolver la funcionalidad usando la sobrecarga definida.

Reglas de negocio generales:

- Todas las funcionalidades deben resolverse con los Gestores definidos. Tiene **prohibido** sacar sublistas o subarreglos para procesar por fuera de los métodos definidos en los Gestores.
- El manejo del dato fecha y el dato hora debe hacerse como un string.