

Programación 4

EXAMEN DICIEMBRE 2024

Por favor siga las siguientes indicaciones:

- Escriba con lápiz y de un solo lado de las hojas
- Escriba su nombre y número de documento en todas las hojas que entregue
- Numere las hojas e indique el total de hojas en la primera de ellas
- Recuerde entregar su número de examen
- Está prohibido el uso de computadoras, tabletas o teléfonos durante el examen

Problema 1 (30 puntos)

Se desea crear un sistema para registrar canales de streaming que ofrecen contenido en línea. De cada canal se conoce su nombre y país de origen. Un canal puede contar con múltiples programas, los cuales tienen un nombre, una calificación global (un rating del 1 al 5) y pueden ser películas, documentales o series. Los programas pueden estar en uno o más canales de streaming. Las series se dividen en episodios, que representan capítulos individuales (numerados consecutivamente) con sus propios títulos y duraciones. Por su lado, las películas tienen su año de lanzamiento y género, que por ahora puede ser: acción, romance o drama. Mientras tanto, de los documentales se conoce su duración en minutos. Los usuarios, por su parte, son quienes consumen estos contenidos. De ellos se conoce su nombre y edad. Los usuarios pueden suscribirse a diversos canales y reproducir programas de los canales a los cuales están suscriptos y, si quieren, también calificarlos. La calificación de un programa por parte de un usuario consta de una calificación numérica, también del 1 al 5, y, si lo desea el usuario, de un comentario en texto. Un usuario solo puede reproducir programas de canales a los que está suscrito y calificar programas que ha reproducido.

Además, se cuenta con la descripción del siguiente Caso de Uso:

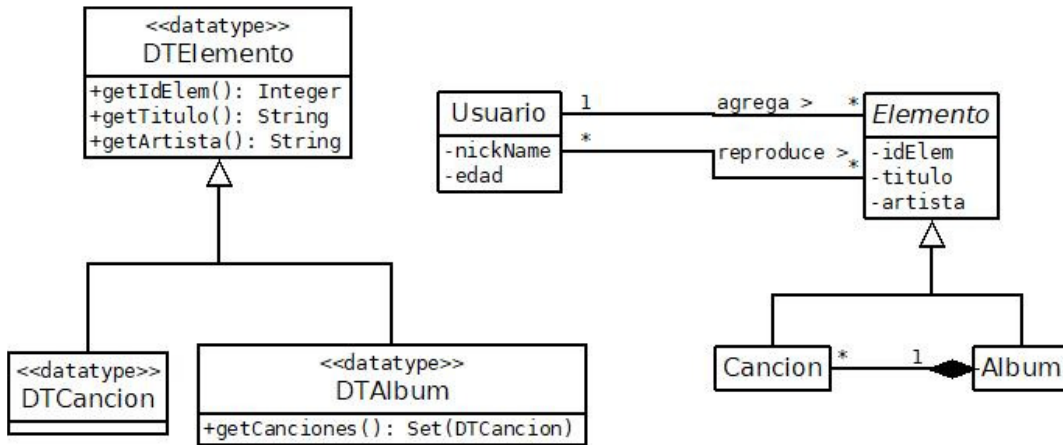
Caso de Uso	Agregar contenido
Actor	Administrador
Descripción	El caso de uso comienza cuando un administrador desea agregar contenido en un canal de streaming. Primero, el administrador ingresa el nombre del canal de streaming y el nombre del contenido a cargar. En el caso de que se quiera ingresar una serie, el administrador ingresa de a uno los datos de los episodios de la serie; para esto ingresa número de episodio, título y duración. Si se trata de una película se ingresan su año de lanzamiento y género. Si se quiere ingresar un documental se ingresa su duración. Para finalizar, el administrador confirma o cancela el ingreso del contenido.

Se pide:

- Realizar el Modelo de Dominio de la realidad planteada, incluyendo todas las restricciones que considere necesarias en lenguaje natural.
- Realizar un Diagrama de Secuencia del Sistema (DSS) para el Caso de Uso. Indique el uso de memoria del Sistema y de datatypes, si corresponde.

Problema 2 (35 puntos)

La figura muestra el Modelo de Dominio de un sistema colaborativo de reproducción de música digital, donde los usuarios pueden agregar contenido (canciones y álbumes) y reproducirlo. Un elemento puede ser una canción o un álbum, que a su vez está formado por canciones. Un usuario puede reproducir contenidos de cualquier otro usuario, incluyendo a sí mismo. Una vez que un usuario reproduce un contenido, se mantiene la asociación entre ese usuario y ese contenido.



Además, se tienen los contratos de las siguientes operaciones del sistema:

agregarNuevoContenido(nickU:string; dtE:DTElemento)	
Descripción	Agrega un nuevo elemento (canción o álbum) al sistema, por parte de un usuario
Parámetros	<ul style="list-style-type: none"> - nickU: Nickname del usuario - dtE: Datos del elemento (canción o álbum) a agregar
Precondiciones	<ul style="list-style-type: none"> - Existe en el sistema una instancia u de Usuario con u.nickName=nickU - No existe en el sistema ninguna instancia e de Elemento tal que e.idElem=dtE.getIdElem() - Si dtE es de tipo DTAlbum, para cada DTCancion c de dtE.getCanciones(), no existe en el sistema ninguna instancia e de Cancion tal que e.idElem=c.getIdElem()
Postcondiciones	<ul style="list-style-type: none"> - Existe un nuevo elemento e en el sistema tal que e.idElem=dtE.getIdElem(), e.titulo=dtE.getTitulo() y e.artista=dtE.getArtista() - Se crea un link de agrega entre la instancia u de Usuario con u.nickName=nickU y el nuevo Elemento e - Si dtE es de tipo DTAlbum, para cada DTCancion c de dtE.getCanciones(), existe una nueva Cancion e' en el sistema tal que e'.idElem=c.getIdElem(), e'.titulo=c.getTitulo() y e'.artista=c.getArtista(), se crea un link entre e' y e, y se crea un link de agrega entre la instancia u de Usuario con u.nickName=nickU y la nueva Cancion e'

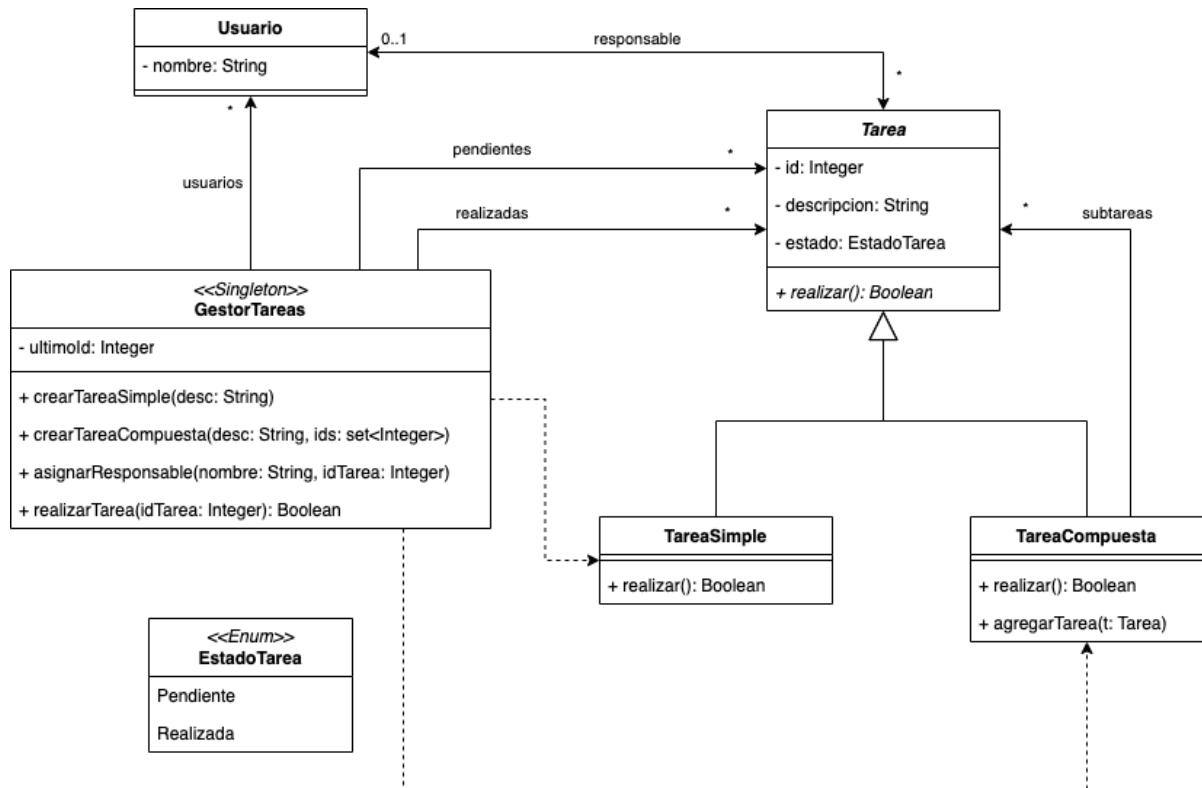
masReproducidoPorEdad(edU: Integer) : Integer	
Descripción	Devuelve el identificador del elemento más reproducido por usuarios mayores que una edad dada
Parámetros	- edU: Edad para filtro de la búsqueda
Precondiciones	No tiene
Postcondiciones	<ul style="list-style-type: none"> - Si no hay en el sistema ninguna instancia <i>u</i> de Usuario tal que $u.edad > edU$ asociada a través de reproduce a ninguna instancia de Elemento, se devuelve -1 - En caso contrario, el resultado <i>r</i> es tal que existe un Elemento <i>e</i> con $e.idElem=r$, <i>e</i> está asociado a través de reproduce con una cantidad <i>M</i> de instancias <i>u</i> de Usuario con $u.edad > edU$ y no existe otra instancia <i>e'</i> de Elemento que esté asociado a través de reproduce con una cantidad <i>M'</i> de instancias <i>u'</i> de Usuario con $u'.edad > edU$, y $M' > M$.

Se pide:

- Realizar los Diagramas de Comunicación para las dos operaciones especificadas.
- Realizar el Diagrama de Clases de Diseño resultante, incluyendo controlador.

Problema 3 (35 puntos)

Se está construyendo un sistema de gestión de tareas que permite administrar tareas simples y compuestas (tareas que contienen otras tareas). El diseño se muestra en la siguiente figura.



Las tareas del sistema pueden estar en dos estados: pendiente o realizada. Cada tarea tiene un identificador único asignado por el sistema (comenzando en 1), una descripción y opcionalmente un usuario responsable de que se realice la tarea.

El sistema mantiene dos colecciones de tareas: una para las tareas pendientes y otra para las tareas realizadas. La clase `GestorTareas` tiene los siguientes comportamientos:

- `crearTareaSimple`: Crea una nueva tarea simple con la descripción recibida, asignándole un identificador único y la agrega a la colección de tareas pendientes.
- `crearTareaCompuesta`: Crea una nueva tarea compuesta con la descripción y la colección de identificadores de tareas recibidas, donde los identificadores de tareas corresponden a tareas pendientes ya existentes en el sistema. Del mismo modo, le asigna un identificador único y la agrega a la colección de tareas pendientes.
- `asignarResponsable`: Asigna un usuario como responsable de una tarea, estableciendo la relación bidireccional entre ambos.
- `realizarTarea`: Recibe un identificador de una tarea de la colección de tareas pendientes e intenta marcar como realizada la tarea identificada. Para tareas compuestas, esto solo es posible si todas sus subtareas ya están realizadas. En caso de éxito, mueve la tarea de la colección de pendientes a la de realizadas, cambia el estado de Pendiente a Realizada y devuelve true. En otro caso, devuelve false y no hace nada más. Para el caso de tareas simples siempre es posible realizar la tarea.

La clase abstracta `Tarea` define el comportamiento común para todas las tareas del sistema, mientras que las clases `TareaSimple` y `TareaCompuesta` implementan los comportamientos específicos para cada tipo de tarea.

Se pide:

- a) Nombrar el patrón de diseño utilizado en la solución, identificando las clases e interfaces participantes y sus roles.
- b) Implementar (.h y .cpp) las clases `Tarea`, `Tarea Compuesta` y `GestorTareas`. Incluir los constructores necesarios.

Observaciones: Utilice colecciones paramétricas (contenedores STL). No incluya directivas al precompilador (`#include`, etc).