****

Back End I

| Durante esta clase, te proponemos realizar las ejercitaciones de las clases anteriores que no llegaste a completar para poder ponerte al día. En caso de haberlas completado, te dejamos los siguientes desafíos para seguir poniendo en práctica lo aprendido. |
| --- |

Clase integradora: patrón proxy

# Objetivo

Realizar el diagrama UML y programar en Java, implementando patrón proxy según el siguiente enunciado. A tener en cuenta:

* Ejercicio individual
* Nivel de complejidad: intermedio 🔥🔥

## Enunciado

Ahora nuestra aplicación de streaming va a incorporar la emisión de series. En este caso, necesitamos desarrollar una funcionalidad en particular.

Como cliente, queremos que la aplicación nos permita hacer un pedido con el nombre de la serie y nos envíe el link de reproducción. El único límite impuesto a nuestra petición es que la cantidad de series no puede ser más de 5 por semana. Para poder saber cuántas ya fueron vistas se llevará un contador.

Se propone crear una estructuras de clases similar al ejercicio anterior, con las siguientes clases:

* Una interfaz ISerie que define un método getPelicula que recibe el nombre, devolviendo un string.
* Una clase Serie que implementa la interfaz ISerie y devuelve un string que representa el link y simplemente es el prefijo “www.”, seguido por el nombre de la serie.
* Una clase SerieProxy con un atributo cantVistas que cuenta las veces que fue pedida una serie y le pide a Serie un link con getPelicula y verifica las veces que se reprodujo alguna serie. Si se supera el valor máximo, lanza una excepción del tipo SerieNoHabilitadaException, creada para tal fin.

En el método main, realizar pedidos de series con el nombre. Mostrar el link de la serie si está habilitada. Si no, mostrar un cartel de “Ha superado la cantidad de reproducciones permitidas”.

Codificar los casos de pruebas que sean requeridos para garantizar la calidad del método getPelicula().

¡Muchos éxitos!