

Introducción: Información general del proyecto

Título: Series-y-Películas-de-Netflix

Repositorio utilizado para desarrollar el análisis del patrón de diseño de un proyecto de POO:

<https://github.com/daliaydom/Series-y-Pelicuas-de-Netflix.git>

Por: Dalia Yvette Domínguez Jiménez

El proyecto seleccionado para el análisis del patrón de diseño se centra en la implementación de la programación orientada a objetos en Java para la creación y gestión de objetos relacionados con películas y series de Netflix. La finalidad principal de esta aplicación es almacenar información relevante sobre cada producción, como el título, creador, género, año de lanzamiento, duración, entre otros.

En el núcleo del proyecto se encuentra el archivo ejecutable.java, que actúa como el punto de entrada principal. Este archivo crea arrays de Películas y Series, utilizando distintos constructores para generar objetos en cada posición. Además, realiza acciones como marcar visualizadas algunas producciones, mostrar listas de las películas y series visualizadas, y calcular la serie con más temporadas y la película más reciente.

La estructura del proyecto se organiza en torno a una interfaz llamada Visualizable, que define métodos como marcarVisto(), esVisto(), y tiempoVisto(). Las clases principales Serie y Pelicula heredan de la superclase Netflix, que contiene getters, setters y el método toString(), el cual es sobrescrito por sus subclases para presentar información específica de cada tipo de producción.

La interfaz Visualizable juega un papel crucial al establecer un estándar para la visualización y marcado de elementos, proporcionando una estructura uniforme para ambas películas y series. En este contexto, el propósito del documento analítico es explorar cómo se ha implementado y aplicado el patrón de diseño, evaluando su efectividad y relevancia en el diseño de la aplicación. Se examinarán las ventajas y desventajas de la elección del patrón, así como posibles alternativas para resolver los problemas específicos abordados en el proyecto.

Complejidad del Diseño en construcción:

1. Manejo de Arrays Dinámicos:

La creación y gestión de arrays dinámicos para almacenar películas y series implica consideraciones sobre cómo gestionar el tamaño de estos arrays de manera eficiente y cómo manejar la adición de nuevos elementos sin comprometer el rendimiento.

2. Eficiencia en el Cálculo de Datos:

El cálculo de la película más reciente y la serie con más temporadas puede requerir un diseño eficiente para procesar y analizar grandes conjuntos de datos, especialmente a medida que el número de elementos en los arrays crece.

3. Gestión de Datos Complejos:

El manejo de información detallada sobre películas y series, que incluye atributos como título, creador, género, año, duración, entre otros, puede ser complejo. Diseñar una estructura de datos eficiente y fácil de mantener para gestionar esta información es crucial.

Patrón de Diseño de Software

El patrón que parece estar presente en este proyecto es el Template Method. Este es un patrón de diseño que define la estructura de un algoritmo en una superclase, pero deja que las subclases sobrescriban pasos específicos del algoritmo sin cambiar su estructura. En este caso, las clases Pelicula y Serie son subclases de Netflix, y ambas sobrescriben el método toString(), que es parte del patrón Template Method.

Para demostrar la pasada deducción, se presentan las clases relevantes que comprueban el patrón de diseño bajo el que opera.

1. Clase Netflix (Super Clase):

Template Method (toString()): La clase Netflix define el método toString(), que actúa como el Template Method. Este método proporciona la estructura general para representar información sobre un video de Netflix.

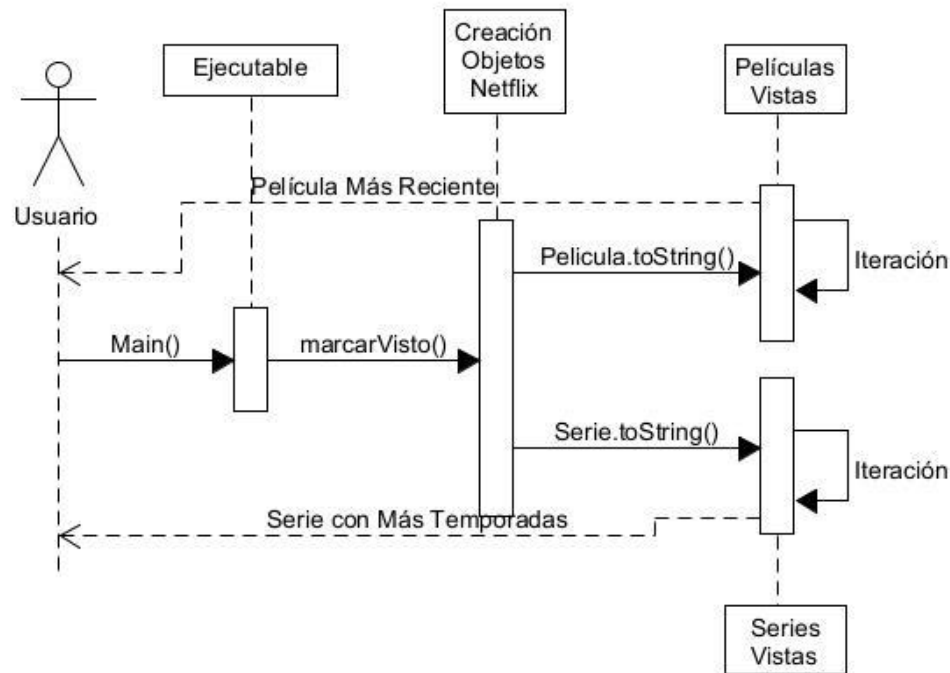
2. Clase Película (Subclase de Netflix):

Sobrescritura del Template Method: La clase Pelicula sobrescribe el método toString() de la superclase Netflix, proporcionando una implementación específica para representar películas. La sobrescritura de este método es una aplicación del patrón Template Method.

3. Clase Serie (Subclase de Netflix):

Sobrescritura del Template Method: Al igual que la clase Pelicula, la clase Serie sobrescribe el método toString() de la superclase Netflix, proporcionando una implementación específica para representar series. Esto también demuestra el uso del patrón Template Method.

Para sustentar lo presentado anteriormente se puede apreciar el diagrama de secuencia:



En este diagrama de secuencia, el patrón Template Method se manifiesta en la ejecución del método `toString()`, donde la superclase define la estructura general y las subclases la completan con detalles específicos. Este enfoque permite que el método `toString()` sea aplicado de manera consistente a través de las diferentes subclases de Netflix.

Contextualización del Uso del Template Method:

1. Estructura Común

La superclase Netflix establece una estructura común al definir el método `toString()`. Este método se utiliza para representar la información general de cualquier video de Netflix.

2. Sobreescritura en Subclases

Las subclases Película y Serie sobrescriben el método `toString()`, proporcionando implementaciones específicas para representar películas y series respectivamente.

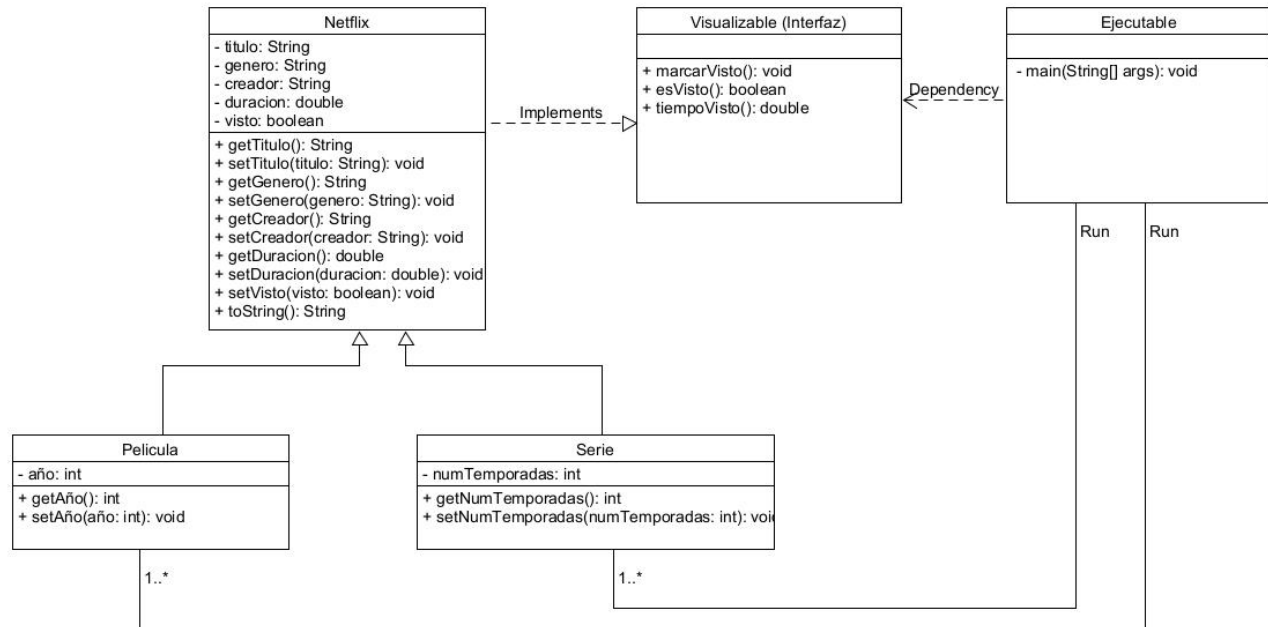
3. Reutilización de Código

El método `toString()` en Netflix actúa como el algoritmo general para representar un video, y las subclases personalizan cómo se presenta esa información específica para películas o series. El patrón Template Method se utiliza para definir la estructura general de representación de videos en la superclase Netflix, permitiendo que las subclases proporcionen implementaciones específicas según el tipo.

Patrón Aplicado al Proyecto

En este proyecto, tiene sentido utilizar el patrón Template Method en el método toString() de la clase Netflix por varias razones, entre estas, la estructura común; la superclase Netflix define la estructura general para representar información sobre un video, ya sea una película o una serie. Al utilizar el patrón Template Method en el método toString(), se establece una estructura común que comparten todas las subclases. Esto promueve la consistencia en la presentación de la información y evita la duplicación de código.

Para comprender lo dicho anteriormente con mayor claridad se puede apreciar el diagrama de clases del proyecto:



En el diagrama de clases, se observa la implementación del patrón Template Method a través de la estructura y comportamientos de las clases Netflix, Película, y Serie, en este punto debe de considerarse el factor de reutilizar código, porque el patrón Template Method permite reutilizar el código base en la superclase (Netflix) mientras proporciona flexibilidad para que las subclases (Película y Serie) personalicen partes específicas según sus necesidades. En este caso, la información general sobre un video (título, creador, género, duración) se presenta de manera consistente en todos los casos, mientras que las subclases pueden agregar detalles específicos como el año de una película o el número de temporadas de una serie.

Conclusiones

La elección del patrón Template Method en este proyecto tiene sentido porque proporciona una estructura común para representar información de películas y series en la clase base Netflix, permitiendo que las subclases Pelicula y Serie personalicen la presentación según sus características específicas. Esto promueve la reutilización de código al definir la lógica general en la superclase y permite una fácil extensión al agregar nuevos tipos de videos en el futuro. Las ventajas incluyen una mayor cohesión y mantenibilidad del código. Sin embargo, una desventaja podría ser la rigidez en la estructura predefinida, ya que cualquier cambio en el Template Method afectaría a todas las subclases. Alternativamente, podrían haberse utilizado patrones de diseño creacionales como Factory Method o Abstract Factory para la creación de instancias de películas y series, brindando una mayor flexibilidad en la creación de objetos y adaptándose a posibles variaciones en la estructura de las clases.

Referencias

- Domínguez Jiménez, Dalia Yvette (Daliaydom). 2021. Series-y-Películas-de-Netflix: <https://github.com/daliaydom/Series-y-Pelicuas-de-Netflix/tree/main> ingresado el 20/11/2023