

REPORTE VIDEO CLUB PROGRAMACIÓN AVANZADA

Victor Huerta
Davor Serey
Ignacio Silva

Profesor:
Claudio Cubillos

Valparaíso
30 de Septiembre de 2022

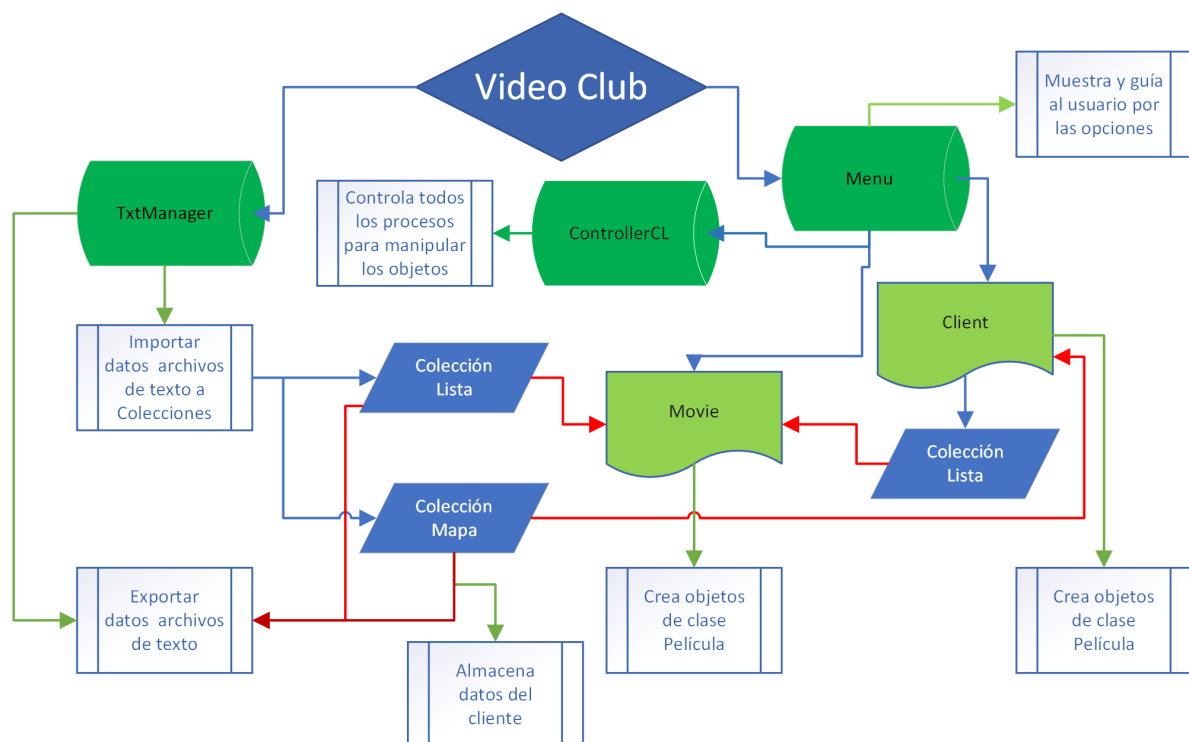
1.1

Para este proyecto, simularemos en una aplicación, un VideoClub para rentar películas, donde haremos uso de un dataframe que contiene una gran cantidad de **películas**, que se conforman por su **nombre**, **distribuidor** y **fecha** de salida de la película. Además se agregaron nuevas categorías para poder implementar las distintas funcionalidades de un videoclub, tales como un **ID**, para facilitar la búsqueda, la **cantidad de veces que se ha arrendado**, y el **rut de la persona actual que la ha arrendado**, cabe recalcar que solamente existe una sola unidad para cada película, esto quiere decir que si la película se encuentra rentada, no está disponible para arrendar hasta que sea devuelta.

No es posible crear nuevas películas, pero con la cantidad con la que se trabaja es suficiente para buscar películas específicas por su nombre o ID respectivo, mostrar todo el catálogo de películas y también se puede encontrar la más popular, es decir, la que más veces ha sido arrendada.

Es posible manipular datos de **clientes**, donde se guardará el **nombre**, **rut** y una **lista de películas** que tiene rentadas actualmente. Los clientes se podrán registrar dentro de la aplicación, se podrán buscar según su rut, también mostrar sus películas. Pueden arrendar y devolver películas a su antojo, mientras estén disponibles.

1.2



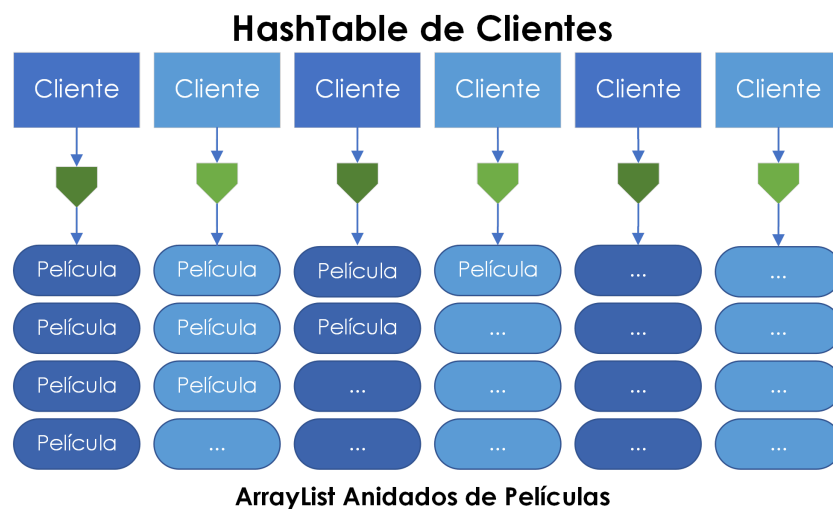
1.3

En este proyecto, todos los atributos de las clases **Movie** y **Client**, son privados, esto quiere decir que sólo serán accesibles sólo desde la misma clase o sus métodos respectivos, tales como getters y setters.

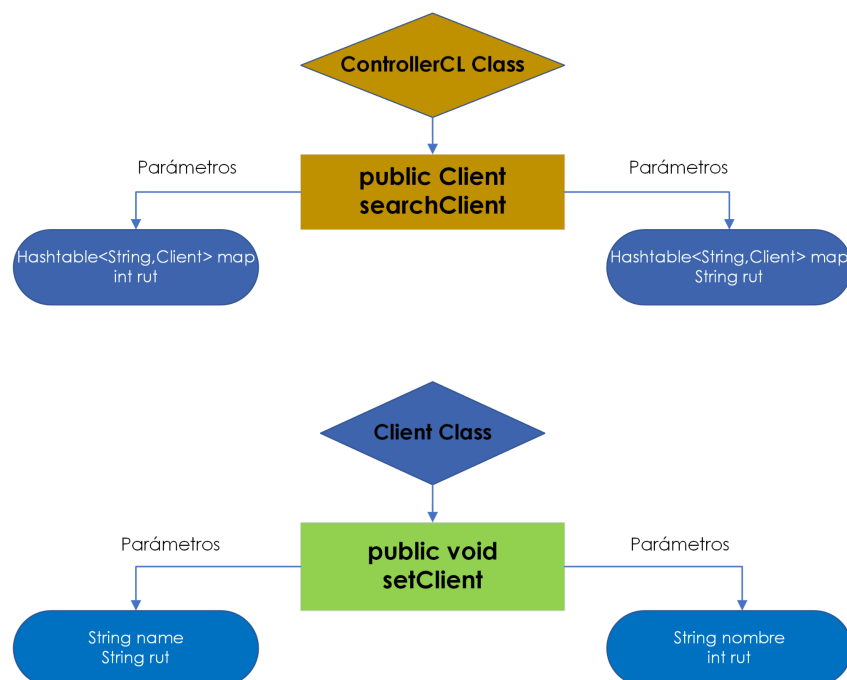
1.4

En el código del programa se implementó la funcionalidad de importar archivos de textos llenos de datos ya mencionados anteriormente. De esta manera se obtienen datos iniciales para su posterior uso durante la ejecución.

1.5

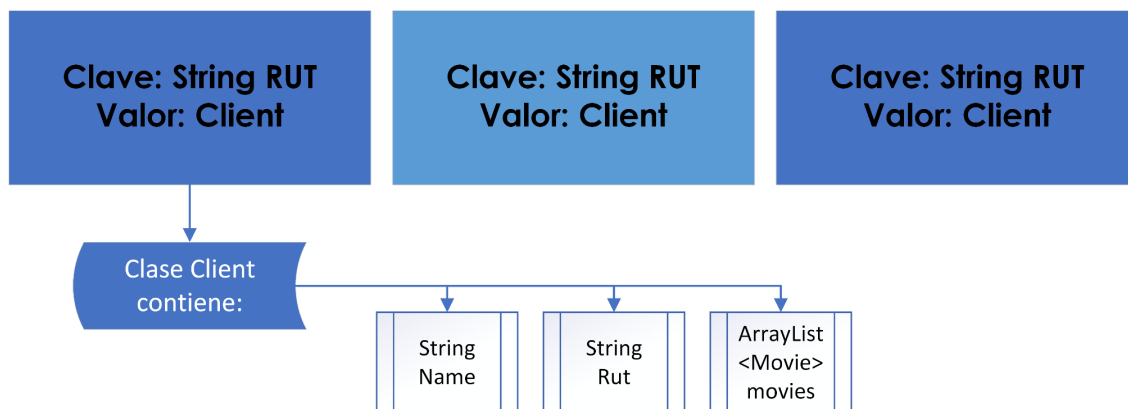


1.6



1.7

Colección Utilizada: Hashtable <String , Client>



1.8

En este programa, se llevaron a cabo la creación de distintas funciones para la clase **Client**, entre estas se encuentran las que están dirigidas a la colección anidada, la cuál es el ArrayList de películas.

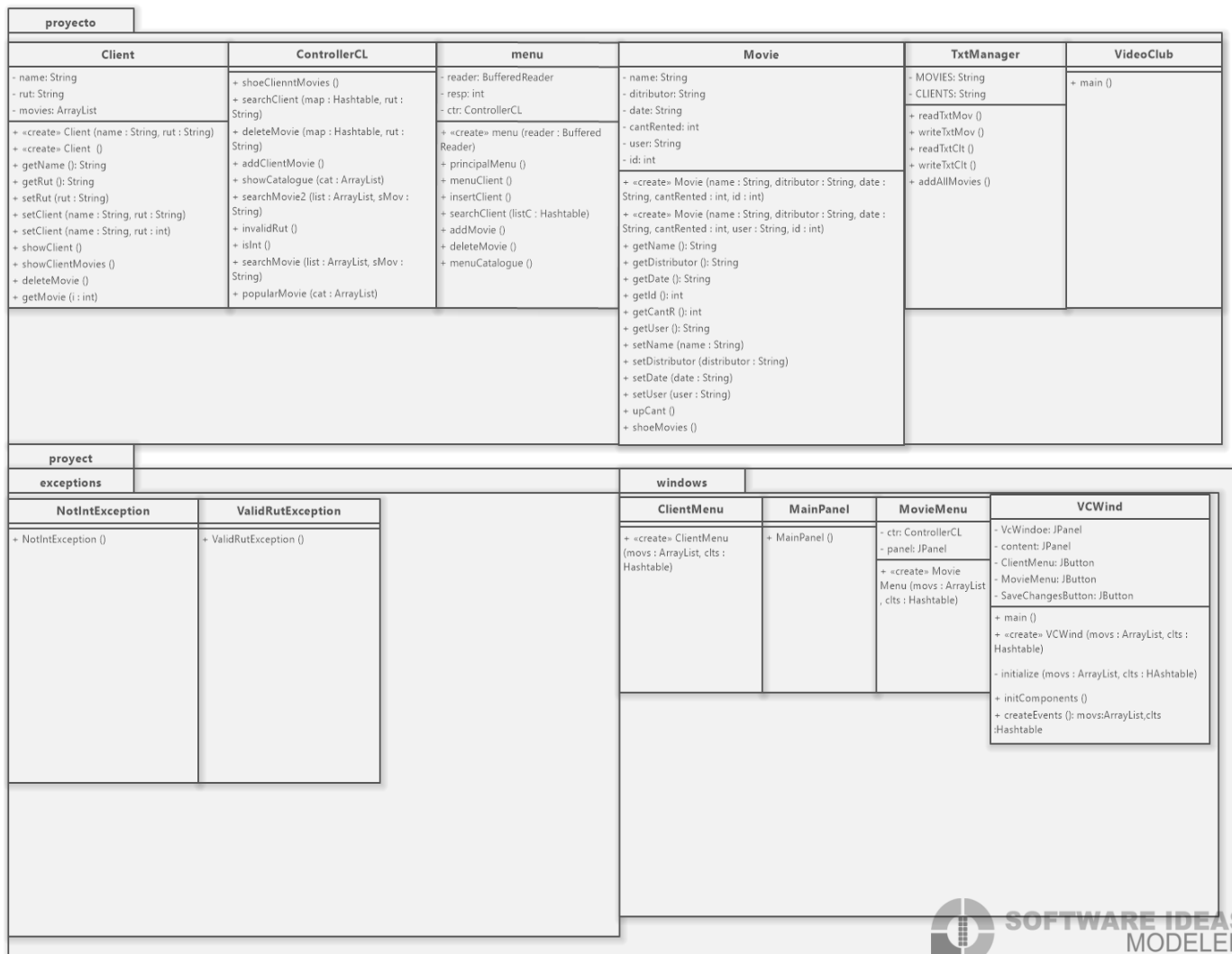
Para esta segunda colección se puede:

- Agregar películas dentro de la lista de películas rentadas de cada cliente.
- También se pueden mostrar todos los datos de los objetos de clase **Movie** contenidas dentro de la lista de películas arrendadas de cada cliente.

Todo lo mencionado anteriormente da la posibilidad de poder ejecutar correctamente el menú ya creado que contiene las diversas funcionalidades necesarias para simular un videoclub.

2.1

Diagrama UML



2.2

Durante la ejecución del programa se importan archivos de texto (extensión .txt) y se trabajan con los datos obtenidos de estos. Mientras se manipulan los datos, se necesitan guardar de alguna manera y tener una mantención de datos para guardar estos cambios. En este proyecto ocurre al finalizar el programa. El proceso es el siguiente:

- Se eliminan los archivos de textos presentes en la carpeta del proyecto.
- Se crean nuevamente archivos de textos, pero vacíos.
- Se ingresan los datos que deseamos guardar dentro de los nuevos archivos.
- Finalmente se cierra el programa con los cambios realizados y guardados.

2.3

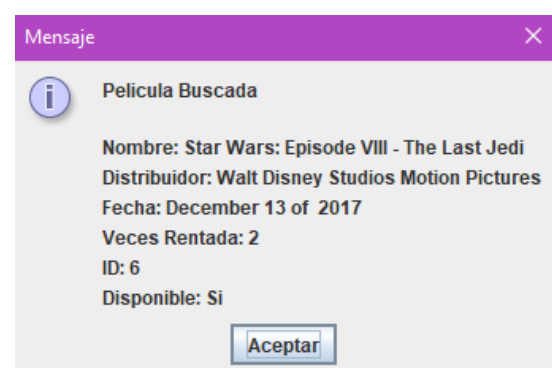
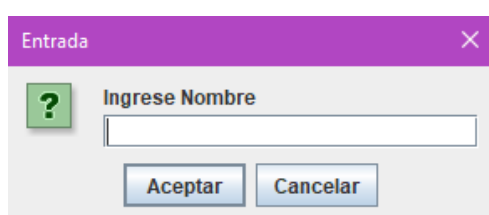
Se han implementado nuevas ventanas para interactuar con el usuario de manera más amena e intuitiva, mostrando los menús donde se pueden hacer las funciones necesarias para el videoclub. Los botones que se encuentran en la esquina superior izquierda sirven para entrar a los diferentes menús que están implementados en las ventanas del programa.



Cuando se desea ingresar datos, se muestran ventanas con campos de texto donde se debe escribir lo que se pide para el correcto funcionamiento del programa.

Se mostrará también el resultado, dependiendo de la función que se haya escogido.

Es necesario aclarar que la única forma de guardar los cambios (exportar) dentro de la ventana es utilizando el botón de la esquina superior derecha (Guardar Cambios).



2.4

Se crean nuevas funcionalidades para la segunda colección anidada, la cuál corresponde al ArrayList dentro de cada cliente. En el transcurso del desarrollo del proyecto, se han implementado las siguientes opciones para el correcto funcionamiento del programa:

- **Edición:** Se puede arrendar una película, esto hace que suba el contador de la cantidad de veces que se ha arrendado esta película, modificando este atributo para su futuro uso.
- **Eliminación:** Se permite a los clientes devolver películas rentadas anteriormente para que se agreguen al catálogo nuevamente y sea posible rentarlas nuevamente.

2.5

Como se mencionó anteriormente se tiene un contador dentro de cada película que aumenta cada vez que se arrienda, esta funcionalidad sirve para poder encontrar las películas más populares. Existe una opción dentro del menú del catálogo que muestra la película más popular, es decir, la película que más veces se ha arrendado. Esto sirve como sugerencia para los clientes que busquen nuevas películas que quizás no hayan visto antes.

2.8

Se implementaron excepciones para su uso y manejo, con el fin de ser capaces de afrontar un error de ejecución y seguir trabajando con el programa, de forma que no se produzca un error fatal, que nos perjudique.

Se presentan especialmente al hacer casteos de objetos y variables, también leer lo que escribe el usuario, o simplemente evitar que se produzcan errores al trabajar con objetos nulos.

2.9

Se crearon dos clases que se extienden de Exception, estas dos son las siguientes:

- **NotIntException:** Esta excepción es arrojada cuando se desea convertir un String a un int directamente. Esta clase es usada al recibir un rut como String y comprobar si está compuesto por números.
- **ValidRutException:** Es usada al comprobar si un rut ingresado es válido para poder utilizarlo (Cantidad de dígitos mayor a 6 y menor a 9).

En general, estas excepciones son utilizadas para validar los ruts para la correcta utilización de los Strings entregados.

2.10

Durante el desarrollo de este proyecto, se ha utilizado GitHub para el control de versiones del programa. Se han hecho múltiples Commits, reportando y subiendo los cambios hechos para el progreso del videoclub.

A lo largo de la entrega del avance, se hicieron en total 5 commits hechos por los integrantes del proyecto. Estos cambios subidos son principalmente actualizaciones de código y subida del reporte.

En el proceso de la entrega final, se implementaron muchos más cambios, en total 21 commits adicionales. La gran mayoría de estas versiones fueron hechas debido a cambios de código para el desarrollo del proyecto del videoclub (Posibles futuras versiones).

Link Repositorio: <https://github.com/Twistos-Team/VideoClub.git>