

ESCUELA DE INFORMÁTICA
PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

REPORTE AVANCE PROYECTO SIA

Administrador de Películas-Clientes en un video club

Participantes

Victor Huerta

Davor Serey

Ignacio Silva

2 de septiembre, 2022

IDEA DEL PROYECTO

Vamos a simular en **java** un Video Club en nuestro programa, en el cual hay un catálogo de películas y un registro de clientes, donde cada cliente tendrá su lista de películas rentadas.

Aclaración: Este proyecto se encuentra en desarrollo, por lo que este reporte y algunos detalles de este trabajo están sujetos a cambios.

CLASES Y COLECCIONES A UTILIZAR

En la implementación se utilizarán cuatro Clases, con sus atributos respectivos (todos privados):

- ❖ **VideoClub:** Corresponde a la clase principal, donde se encuentra la función *Main*, esta derivará a la siguiente clase.
- ❖ **Menu:** En esta clase se controlarán todas las funcionalidades del programa, guiando al usuario por las distintas herramientas que están a su disposición.
 - **BufferedReader reader:** Es el encargado de recibir las respuestas del usuario, leyendo lo que se escribe en el teclado.
 - **Int resp:** Es el atributo que guarda la respuesta del usuario para poder ingresar al submenú deseado.
- ❖ **Movie:** Guardará todos los datos de las películas.
 - **String name:** Guarda el nombre de la película.
 - **String director:** Es donde se encuentra el nombre del director que dirigió la película.
 - **Int id:** Este es el ID de la película, si se conoce, se puede buscar y encontrar más fácil.
- ❖ **Client:** Conservará los datos de los clientes registrados.
 - **String name:** Corresponde al nombre del cliente.
 - **String rut:** Es el RUT del cliente registrado, como esta clase se implementó en un Hashtable, el **rut** funciona como **key**.
 - **Int cantMovies:** Cantidad total de películas arrendadas con anterioridad.
 - **ArrayList<Movie> movies:** Es la lista donde se guardarán todas las películas que el cliente registrado rentó alguna vez.

Las colecciones que se implementarán en este proyecto son los siguientes:

- ❖ **ArrayList <Movie> catalogueM:** Esta ArrayList contendrá todas las películas que se pueden rentar dentro del VideoClub, en el cual el **índice** donde se guarda el elemento en la lista corresponde al **ID** de la película.
- ❖ **Hashtable <String, Client> listC:** En este Mapa, se guardarán todos los clientes registrados en el VideoClub, donde la **key** vendría siendo el **RUT** de la persona, y el **value** es un cliente (**Client**).
 - Dentro del Hashtable de Clientes, se encuentra anidado un **ArrayList <Movie> movies**, alojado en los atributos de clase de la clase **Client**, que representa todas las películas que el cliente haya rentado anteriormente.

PRINCIPALES FUNCIONALIDADES

Con la clase **Menu**, el usuario podrá acceder a las diferentes opciones que tendrá disponible, con las cuales podrá manipular todo el programa a su conveniencia. En todos los menús, si no se ingresa una opción correcta, se saldrá del menú actual.

Se empezará con el **menú principal** donde se mostrarán las siguientes opciones:

- ❖ **Menu de clientes:** Se encontrarán todas las funcionalidades relacionadas con los datos de los clientes registrados en el VideoClub.
 - **Registrarse como cliente:** El usuario, si no se encuentra ya en el sistema, se podrá registrar y rentar películas en un futuro.
 - **Mostrar cliente:** Se busca por RUT el cliente, y se muestra la información del cliente buscado, si no se encuentra, se mostrará en pantalla “Cliente no existente”.
 - **Arrendar película:** Se ingresará la película deseada al ArrayList contenida en los atributos de clase del cliente.
- ❖ **Menu de películas:** En este apartado, se encontrarán todas las funciones para manipular las películas guardadas en el catálogo del VideoClub.
 - **Mostrar todas las películas:** Se mostrarán en pantalla todas las películas presentes en el catálogo de películas.
 - **Buscar película por nombre:** Función no disponible en programa.
 - **Buscar película por ID:** Función no disponible en programa.

METODOS Y CONSTRUCTORES

Veremos todos los métodos y constructores de cada clase en el proyecto VideoClub.

Clase VideoClub

- ❖ **Public static void main (String):** Donde se crean las colecciones, donde se llama a la función que lee los archivos .txt de películas y se crea una variable tipo **Menu** para llamar al menú principal.
- ❖ **Public static void readTxt (ArrayList<Movie>):** Este método es el encargado de leer los archivos .txt y guardar los datos dentro de la ArrayList de películas.

Clase Menu (Un solo constructor, sin sobrecarga de métodos)

- ❖ **Public Menu ():** Constructor correspondiente a la clase **Menu**, se inicializa **resp = 0**, y **reader** como un lector de inputs.
- ❖ **Public void principalMenu (ArrayList<Movie>, Hashtable<String, Client>):** Corresponde al menú principal explicado anteriormente, que deriva a los otros submenús.
- ❖ **Public void menuClient (ArrayList<Movie>, Hashtable<String, Client>):** Es donde se encuentran las opciones disponibles para el menú de clientes.

- ❖ **Public insertClient (Hashtable<String, Client>):** Se leen los datos del usuario que desea registrarse y los inserta en el **Hashtable** de clientes, con la **key** como **RUT**, y el **valor** como la clase **Client**.
- ❖ **Public searchClient (Hashtable<String, Client>):** Se ingresa un RUT, y se busca el cliente con el rut correspondiente, finalmente retorna la clase **Client** (el buscado o null).
- ❖ **Public void addMovie (ArrayList<Movie>, Hashtable<String, Client>):** A un cliente en específico, se le insertará la película buscada por su nombre mediante un método **Client**. Finalmente mostrará todas las películas que el cliente tenga en su registro.
- ❖ **Public void menuCatalogue (ArrayList<Movie>, Hashtable<String, Client>):** Se mostrarán y se derivarán a los siguientes submenús relacionados con la clase **Movie**.
- ❖ **Public void showCatalogue (ArrayList<Movie>):** Mostrará todas las películas que se encuentran en el catálogo del VideoClub.

Clase Client (Sobrecarga de constructores, sobrecarga de métodos)

- ❖ **Public Client ():** Constructor que se no se le entrega nada. Se inicializa el nombre y el rut en null, la cantidad de películas en 0 y la lista de sus películas en null.
- ❖ **Public Client (String, String):** Constructor que como parámetros tiene un String correspondiente al nombre del cliente, y el rut como String. Se inicializa la cantidad de películas en 0 y la lista de sus películas en null.
- ❖ **Public setClient (String, String):** Se le da como valor los parámetros String a los atributos nombre y rut.
- ❖ **Public setClient (String, int):** Como parámetros se le entrega un String nombre, y un entero, que se le inserta al rut casteado como String.
- ❖ **Public showClient ():** Se muestran los datos del cliente.
- ❖ **Public setClientMovie (Movie):** Se le agrega la película que se entrega a la lista de películas del cliente.
- ❖ **Public showClientMovie ():** Encargado de mostrar todos los elementos de la lista de películas del cliente.

Clase Movie

- ❖ **Public Movie ():** Sin parámetros, se inicializa a 0 o null todos los atributos.
- ❖ **Public Movie (String, String, int):** Se darán como valor a los atributos, nombre como String, director como String y el ID como entero.
- ❖ **Public String getName ():** Retorna el nombre de la película.
- ❖ **Public void showMovie ():** Muestra los datos de los atributos de la película.