

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

# Proyecto Estructura de Datos: Your Movie Discovery

Camila Díaz  
Rodrigo Araos  
Vicente Mercado

Profesor: **Ignacio Araya**  
Asignatura: **Estructura de Datos (ICI2240-1)**

**Junio 2022**

# Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Dominio del problema</b>	<b>2</b>
<b>3</b>	<b>Descripción de la aplicación</b>	<b>3</b>
3.1	Funcionalidades de la aplicación	3
3.1.1	Menú de bienvenida	3
3.1.2	Menú de opciones	3
3.1.3	Menú de categorías	4
3.2	¿Qué no puede hacer la aplicación?	5
<b>4</b>	<b>Descripción de la solución</b>	<b>6</b>
4.1	Tipos de Datos	6
4.2	Tipos de Datos Abstractos (TDAs)	7
4.2.1	Mapa	7
4.2.2	Lista enlazada	8
4.3	Implementación	8
4.4	Utilidades extra	10
<b>5</b>	<b>Planificación</b>	<b>11</b>
<b>6</b>	<b>Conclusión</b>	<b>12</b>
<b>7</b>	<b>Coevaluación</b>	<b>13</b>

# 1. Introducción

A día de hoy, el mundo del cine y el séptimo arte se ha vuelto uno muy amplio. Con cerca de 800 producciones siendo estrenadas cada año, y considerando la inmensa variedad de géneros que éste abarca en cada uno de sus estrenos, se puede asegurar que siempre habrá algo para cada persona; desde los niños hasta los más viejos, los que aman la comedia o adoran el terror, el séptimo arte se ha convertido en uno de los medios de entretenimiento más vistos, que tanto los casuales como los más devotos, autodenominados *cinéfilos*, pueden disfrutar.

Sin embargo, como todo arte, la existencia de diferentes puntos de vistas es inevitable. Mientras que existen películas con un consenso compartido entre espectadores casuales y críticos especializados, este no siempre es el caso. Y lo mismo se puede aplicar entre las opiniones de los misma audiencia mencionada anteriormente.

Si bien existen comunidades y sitios web centrados en crear un ambiente para hablar de cine y expresar las opiniones de cada uno, lo cierto es que, con el hecho de que cada vez hay más estrenos cada año, a su vez se vuelve más difícil encontrar y elegir alguna producción que pueda interesar al espectador y sus gustos. En otras palabras, el buscar una película *interesante* de ver se vuelve cada vez en una tarea más ardua, llegando incluso a volverse agobiante para ciertas personas. Y para rematar, en gran parte de sitios las opciones de filtrado, aunque amplias, no siempre son las que se requieren, en especial para casos más aislados tal como una **afinidad entre dos o más usuarios**.

Es así como esto último, además de las "problemáticas" mencionadas anteriormente, entregan la información necesaria que sugiera considerar el desarrollo de un software que permita conectar con más cercanía a los usuarios amantes del cine, siendo los gustos de cada uno más importantes que la opinión generalizada, lo que resultaría en un sistema de recomendaciones más completo y amigable para el usuario, evitando dejarse llevar por algoritmos complejos que solo sugiera películas relacionadas.

## 2. Dominio del problema

El problema que trata de solucionar la aplicación es evitar que el usuario se agobie y esté indeciso ante el amplio rango de películas que se encuentran a su disposición en distintas plataformas de streaming, de las cuales no sabe cuales son afines a sus gustos y cumplen con las expectativas que tiene sobre el contenido que consume. Además, sin la ayuda de un sistema de recomendaciones el espectador malgasta su tiempo tratando de encontrar un filme y puede llegar a ocupar en su totalidad el tiempo que tenía destinado a esta actividad.

La aplicación espera poder agilizar este proceso de selección, en especial a aquellos que no tienen bastante tiempo de ocio, fundándose en usuarios con gustos afines al espectador y recomendándole películas que ellos han visto de las cuales el usuario no se haya percatado. Asimismo, se espera que la aplicación pueda organizar y clasificar las películas del usuario a partir de un criterio en específico, ya sean aquellas que le agradaron, desagradaron o todas las que ha visto, con el fin de recomendar estas películas a otros usuarios con gustos semejantes.

### 3. Descripción de la aplicación

Nuestra aplicación consiste en realizar recomendaciones de películas importadas desde un archivo externo, a partir de las preferencias del usuario en cuestión, de esta forma ayudándole a facilitar la manera de elegir una en específico para ver. Luego, y si el filme le ha gustado, podrá también compartirla con otros usuarios con gustos similares (afines) que aún no lo han visto y viceversa.

#### 3.1. Funcionalidades de la aplicación

A continuación, se presentarán las funciones que el programa será capaz de realizar para el uso correcto de la aplicación por parte del usuario.

##### 3.1.1. Menú de bienvenida

Es el primer menú que el usuario verá al iniciar la aplicación, la cual pedirá un ID o apodo a ingresar. El programa buscará el archivo CSV con dicho dato como nombre dentro de una carpeta dedicada. Si el ID ingresado no existe en el sistema, se crea un nuevo CSV cuyo nombre sea el nuevo dato y se cargará en el programa.

##### 3.1.2. Menú de opciones

Es la interfaz que se visualiza tras la inserción del ID, y contempla las siguientes funciones:

- **Añadir/actualizar películas a la lista:** Al elegir esta opción, se mostrará al usuario dos opciones para subir películas a su archivo:
  - a) *Formato manual:* Al usuario se le dará la posibilidad de insertar los datos de su película conforme el programa los vaya pidiendo (año, rating, genero, etc.). Finalizada la inserción de la película, se consulta si quiere insertar otra película. Si el usuario así lo desea, se retorna al inicio del procedimiento, de lo contrario se retorna al menú principal.
  - b) *Importar desde CSV:* El programa abrirá el archivo 'ratings.csv' que se ubica en la carpeta raíz de la aplicación. Si el programa encuentra el archivo de dicho nombre, se comienzan a importar todas las películas desde allí. Procesadas todas las películas, se retorna al menú principal. En caso contrario, se avisará con un mensaje en pantalla de que no se pudo encontrar el archivo, retornando posteriormente al menú principal.
- **Buscar película por ID:** Se le solicitará al usuario que ingrese el ID de alguna película, para así ser mostrada toda la información de ésta en pantalla, como su nombre, año de estreno y género(s), además de la puntuación promedio entre usuarios que la han visto. En caso de que no exista el filme buscado en la base de datos del programa, se le informará al usuario de aquello.

- **Mostrar películas de un usuario ordenadas por criterio:** Al acceder a esta función, se le preguntará al usuario si desea ver las películas que ha visto, o las de otro usuario en particular. En este último caso, se deberá ingresar el nombre del usuario cuyas películas se van a analizar. Acto seguido, se mostrará un menú con las distintas categorías para mostrar las películas de dicha persona por pantalla (ver la sección 3.1.3 para más información). Una vez seleccionada una categoría, se procederá a mostrar las películas del usuario en cuestión respetando la condición, para luego retornar al menú principal.
- **Recomendar películas a usuarios según afinidad:** Se preguntará por el nombre (ID) del usuario con el cual se quiere comparar gustos. Si este se encuentra dentro del sistema, el programa se encarga de comparar ambas listas de películas e identificará la afinidad entre los usuarios a partir de la compatibilidad de gustos en películas favoritas y las puntuaciones en ellos. Una vez terminado este proceso, se mostrarán por pantalla aquellos filmes favoritos del CSV del usuario comparado que aún no han sido vistas, respetando la afinidad hallada entre ambos sujetos.
- **Buscar usuarios afines:** El programa se encarga de comparar la lista de películas del usuario actual con los existentes dentro del sistema y mostrar por pantalla aquellos usuarios con gustos en películas favoritas compatibles con las del actual junto con dichos filmes.
- **Mostrar perfil de un usuario:** El programa pedirá el ID de un usuario. Una vez ingresada la información y hallado su archivo correspondiente, se mostrará por pantalla los datos de dicho usuario (N° de películas vistas, sus más repudiadas, favoritas, etc.). En caso contrario, se muestra un mensaje pertinente por pantalla y se retorna directamente al menú principal.

### 3.1.3. Menú de categorías

Es el menú al cual se accederá a partir de la función "Mostrar películas de un usuario ordenadas por criterio". Las categorías disponibles son las siguientes:

- **Por género:** El usuario deberá insertar un género de películas, el programa se encargará de mostrar las películas de dicho género.
- **Por año:** El usuario debe insertar un año, el programa mostrará las películas que fueron lanzadas dentro de esa fecha.
- **Por rating:** El programa mostrará todas las películas en orden de las mejor valoradas hacia las peor valoradas, o viceversa.
- **Por título:** El programa mostrará todas las películas ordenadas por título de la A a la Z.
- **Por tiempo de duración:** El programa muestra todas las películas ordenadas desde las más largas hasta las más cortas en minutos, o viceversa.

- **Por orden de lanzamiento:** El programa mostrará todas las películas ordenadas desde las más viejas hasta más nuevas, o viceversa.

### 3.2. ¿Qué no puede hacer la aplicación?

Pese a todo lo mencionado en la sección anterior, la aplicación no será capaz de:

- Reproducir las películas importadas.
- Procesar archivos de formato distinto a CSV.
- Importar películas directamente desde una página web por medio de Internet.
- Actualizar opinión de una película ya importada.
- Procesar datos que no sean películas (series de TV).
- Manejar fechas u horas en tiempo real.

## 4. Descripción de la solución

Para lograr una ejecución correcta y eficiente de la aplicación en cuestión y una buena implementación de las distintas funciones que comprende, se empleará un conjunto de diversos TDAs y Estructuras de Datos.

### 4.1. Tipos de Datos

El programa funciona con las siguientes estructuras de datos, cada una con sus respectivos campos o atributos.

- **movieType:** Estructura que contiene los datos referentes a cada película procesada por el programa. Sus campos son:
  - **name:** Variable de tipo string, es el nombre de la película en inglés
  - **movie\_id:** Variable de tipo entero, corresponde al ID único de la película.
  - **year:** Variable de tipo entero, corresponde al año de estreno del filme.
  - **genres:** Lista que almacena todos los géneros de la película, todos en inglés, como horror, animation, parody, drama, entre otros.
  - **userScore:** Variable de tipo entero, es la puntuación que le entregó el usuario a la película, en una escala de 1 a 10. Además, se define que, si la película tiene un puntaje mayor o igual a 9, ésta será una de las favoritas del usuario, mientras que, si su puntaje es menor o igual a 2, será una de las que repudia.
  - **runtime:** Variable de tipo entero, es la duración de la película en minutos.
- **userType:** Tipo de dato que guarda todo lo referente al usuario y sus datos a utilizar en el programa. Sus campos son:
  - **user\_id:** Variable de tipo string, corresponde al nombre del usuario.
  - **movieMap:** Mapa que guarda todas las películas vistas por el usuario.
  - **movieNumber:** Variable de tipo entero que guardará el conteo de películas que ha visto el usuario.
  - **abcOrder:** Mapa que guarda las películas, pero en orden alfabético.
  - **yearOrder:** Similar al anterior, es un mapa que guarda las películas en orden. En este caso, ordenadas de acuerdo a su año de lanzamiento.
  - **ratingOrder:** Mapa que guarda cada película en el orden de sus puntuaciones dadas por el usuario.
  - **runtimeOrder:** Mapa que guarda las películas por orden de duración en minutos.
  - **moviesByGenre:** Mapa que contiene listas referentes a cada género, que a su vez contienen las películas que sean de aquel género.



- **movieCategory:** Estructura que almacena la información requerida para trabajar en los mapas que contemplan año, puntuación del usuario y duración de la película, con tal de evitar la repetición de claves. Sus campos son:
  - **name:** Variable de tipo string, corresponde al nombre de la categoría como tal.
  - **movie\_list:** Lista que guarda las películas que contienen el valor de la categoría.

## 4.2. Tipos de Datos Abstractos (TDAs)

Dentro de los TDAs presentes en el desarrollo del programa, se encuentran:

### 4.2.1. Mapa

Los mapas *movieMap*, *globalMovieMap*, *usersMap* y *moviesByGenre* funcionarán a partir de tabla hash, mientras que *abcOrder*, *yearOrder*, *ratingOrder* y *runtimeOrder* funcionarán con un árbol binario de búsqueda.

- *movieMap*: Guarda las películas que ha visto el usuario y su información. Se ocupará el ID de una película como la key y un dato de tipo *movieType* (los datos de dicha película) como value. Su tipo de mapa permitirá una inserción y búsqueda rápidas y sencillas, ya que son de complejidad temporal  $O(1)$  gracias a la función hash.
- *globalMovieMap*: Al igual que la anterior, guarda las películas y sus datos correspondientes, con la diferencia de que aquí se almacenarán todos los filmes ingresados independiente del usuario, donde igualmente la key de cada valor será el ID de cada película, guardándose un dato de tipo *movieType*.
- *usersMap*: Se encarga de guardar la información de cada usuario, es decir, datos de tipo *userType*. La clave de cada dato será el nombre de cada uno de los usuarios cargados en el mapa.
- *abcOrder*: Guarda las películas vistas por el usuario, ordenadas alfabéticamente. Se ocupará el nombre de una película como la key y un dato de tipo *movieType* (los datos de dicha película) como value. El treemap ayudará con un proceso más sencillo en el orden de las películas presentes a partir de las distintas keys y el orden alfabético entre ellas.
- *yearOrder*: Guarda las películas vistas por el usuario, ordenadas por el año de lanzamiento. Se ocupará el año de la película como la key y un dato de tipo *movieType* (los datos de dicha película) como value. El treemap ayudará con un proceso más sencillo en el orden de las películas presentes según las distintas fechas de estreno.
- *ratingOrder*: Guarda las películas vistas por el usuario, ordenadas por rating. Se ocupará el rating de la película como la key y un dato de tipo *movieType* (los datos de dicha película) como value. El treemap ayudará con un proceso más sencillo en el orden de las películas presentes según la opinión del usuario en cada caso.

- *runtimeOrder*: Guarda las películas vistas por el usuario, ordenadas por el tiempo de duración en minutos. Se ocupará la duración de la película como la key y un dato de tipo *movieType* (los datos de dicha película) como value. El treemap ayudará con un proceso más sencillo en el orden de las películas presentes según el tiempo de duración en cada caso.
- *moviesByGenre*: Guarda las listas de generos vistas por el usuario, las cuales poseen las películas correspondientes a un determinado genero. Se ocupará un genero de películas como la key y una lista de películas como value. La tabla hash ayudará a una búsqueda de datos rápida y sencilla, ya que su complejidad temporal es  $O(1)$  gracias a la función hash.

#### 4.2.2. Lista enlazada

- *genres*: Lista enlazada doble que almacena todos los géneros de la película, todos en inglés, como horror, animation, parody, drama, entre otros. Esto permitirá una inserción más directa y eficiente, ya que posee una complejidad temporal  $O(1)$ .
- *movie\_list*: Lista enlazada que se encuentra dentro del tipo de dato *movieCategory*. Guarda todas las películas que se encasillan dentro de la categoría correspondiente.

### 4.3. Implementación

A continuación, se explicará el cómo trabajará el programa, enumerando y profundizando en las funciones introducidas anteriormente, junto con otras opciones que proveerá al usuario una experiencia más completa.

- **Login:** Al momento de iniciar el programa, el usuario es recibido con una pantalla de bienvenida, en donde se le solicitará que ingrese su nombre de usuario. La aplicación se encargará de buscar y abrir el archivo .csv ("*nombre de usuario*").csv en la carpeta de 'users'. Si este no existe, con ayuda de la función '*fopen*' se creará un nuevo archivo que almacene los datos del usuario.
- **Cargar datos del usuario:** Luego de abrir el archivo del usuario especificado, se creará un dato *userType* inicializando todos sus datos. En caso de que el usuario sea totalmente nuevo, tan solo se asigna su nombre de usuario, y se lleva el usuario al menú principal. En cambio, en el otro caso con ayuda de la función incluida '*get\_csv\_field*', en donde cada línea corresponde a una película y sus datos. Así, cada filme corresponderá a una nueva variable *movieType* que será inicializada con una función auxiliar, que será guardada en cada mapa, ordenado o no, que contiene *userType*. El nombre de este usuario además se guardará en una variable string llamada '*loggedUser*', para así facilitar el acceso al perfil o datos de éste en otras funciones.
- **Cargar datos de otros usuarios:** Inmediatamente después y de forma automática, se ingresa al directorio *users* en busca de más archivos .csv correspondientes a

otros usuarios. Así, recorriendo cada fichero se repite el proceso anterior de carga de películas con cada uno de ellos.

- **Menú principal:** Tras cargar los datos del usuario, la aplicación mostrará el menú con todas las opciones, o en este caso, las siguientes funciones:
  - **Añadir películas:** Al seleccionar esta opción, la aplicación abrirá un submenú que proporcionará dos formas de agregar nuevos filmes a la aplicación:
    - **Añadir película(s) manualmente:** Ingresando a un ciclo while, al usuario se le pide el ingreso de los datos del filme, que serán guardados en su correspondiente variable tras presionar Enter. Si el programa detecta una ID que ya se encuentra en el mapa de películas del usuario, se le avisará al usuario y se mostrará en pantalla los datos de la película que trató de ingresarse. Al confirmarse todos los datos, se exportará la película al archivo .csv del usuario como tal. Al finalizar con una película, se le consultará al usuario si desea añadir otra, escribiendo Y o N, donde 'Y' significa que el ciclo continua, o 'N' para terminar el proceso y volver al menú principal de la aplicación.
    - **Importar ratings desde IMDb:** Al igual que el inicio de sesión, esta función conlleva el uso de 'get\_csv\_field' para la carga del archivo '*ratings.csv*' proveído por el sitio web IMDb, el cual debería estar ubicado en la carpeta raíz donde se ubica el programa. El programa abre el archivo, y recibe solo los datos necesarios de la película. Además, considerando que el archivo proviene de una base de datos que no solamente recopila películas, se verifica el tipo de medio antes de agregar algo a cada mapa del usuario. Cada película se exporta al archivo del usuario. Al finalizar, se cierra el archivo y se regresa al menú principal.
  - **Buscar película:** Al seleccionar esta opción, se solicitará el ingreso del ID de la película a buscar, con tal de evitar problemas en la comparación de strings y como cada nombre puede variar. El programa se encargará de buscar en cada CSV existente la película señalada. Si es encontrada, se mostrarán los datos del filme, junto con la puntuación promedio de los usuarios que la han visto. En cambio, si no existe, se le indicará al usuario esta información. Cabe recalcar que, como la aplicación no trabaja con una base de datos gigante como aquellas de Internet, la búsqueda de filmes se verá limitada a solo las que han sido agregadas por los usuarios anteriormente.
  - **Mostrar películas vistas por un usuario:** Al usuario se le preguntará si desea mostrar los filmes que su perfil contiene, o las de otra persona en particular, en donde se le solicitará que ingrese el nombre de usuario de aquel perfil. Luego, sin importar la opción, se le preguntará por qué criterio desea que se muestren las películas, y dependiendo de la elección, se ingresa al mapa que contiene las películas ordenadas, ya sea por año, rating, género, título o tiempo de duración, y se muestran en pantalla solo por su título.
  - **Recomendar películas entre usuarios:** Se le preguntará al usuario el nombre de otro con el cual desea ver sus recomendaciones. Si el archivo .csv del

usuario se encuentra en la carpeta 'users', se abre dicho archivo para ser procesado. Si no, se le informa al usuario, y se vuelve al menú principal. Los criterios de comparación entre gustos de usuarios se define en tres partes: películas favoritas que concuerdan, géneros más vistos entre ambos y su puntuación de aquellos. Por ejemplo, en el mejor de los casos, muchas de las películas favoritas del usuario que inició sesión (o usuario A) concuerdan con las del consultado (usuario B), por lo que allí solo bastaría con recomendar las favoritas del usuario B que el usuario A aún no ha visto. Sin embargo, esto puede variar, por lo que, con ayuda de variables auxiliares, se contabilizan los géneros más vistos, y se promedia la puntuación de ellos, para así otorgar unas recomendaciones más acercadas a lo que busca en realidad el usuario. Estas sugerencias mostrarán los datos de cada película.

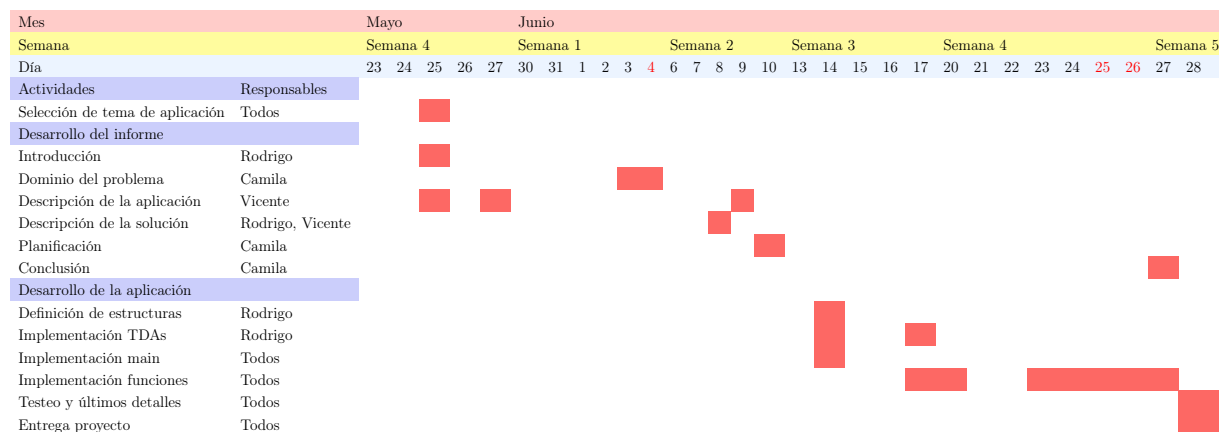
- **Buscar usuarios afines:** Al cargar esta función, el programa se encargará de indagar por los usuarios que compartan más los gustos con aquel que inició sesión, utilizando los mismos criterios anteriores, si no similares, pero con la diferencia de que en esta ocasión se enumerarán en pantalla los nombres de los usuarios encontrados con los mencionados criterios.
- **Mostrar perfil de un usuario:** La aplicación le dará la opción al usuario de mostrar su propio perfil, o buscar el de otro usuario. Sea cual sea la opción, el programa cargará los datos de este usuario si existen, y guardará todo en su respectiva variable 'userType'. Se mostrarán en pantalla su nombre, el número de películas que ha visto, las películas que más repudia o ama y sus géneros más vistos. Tras esto, el usuario puede presionar cualquier tecla para volver al menú principal.

#### 4.4. Utilidades extra

- **Dividir géneros:** Debido a que los géneros están descritos en el csv con el siguiente formato: (*genero1,genero2,genero3,etc*, es necesario implementar esta función que divide en tokens cada uno de los géneros y los añade a la lista de géneros correspondiente a la película.
- **fileExists:** Función que permite verificar si un archivo ya existe para evitar problemas al momento de abrir archivos para editar.
- **Insertar películas:** Función que resume el proceso de inserción de una película a cada mapa disponible en la aplicación, ya sea para el usuario o el mapa global.
- **Exportar películas:** Se recibe el dato de película (movieType) junto con otras variables de tipo string, los cuales se imprimen en el archivo del usuario en la última línea disponible. Los datos se ingresan al archivo en el siguiente orden: ID, nombre, año, géneros, duración, puntuación del usuario.

## 5. Planificación

A continuación, se muestra la planificación realizada por los integrantes del grupo y la distribución de las tareas para el adecuado desarrollo del proyecto.



Cuadro 1: Carta Gantt

## 6. Conclusión

En el trabajo realizado se ha propuesto una aplicación de recomendación de películas a usuarios de plataformas de streaming, la cual lo ayuda con la toma de decisiones frente a amplia variedad de películas que se encuentran disponibles para el cliente. La solución programada ofrece al consumidor la opción de poder simplificar su elección mediante un sistema de afinidad con otros usuarios que comparten sus gustos, con el propósito de encontrar filmes de los cuales no se haya percatado y se ajuste a sus gustos.

A través del desarrollo de este proyecto se ha mantenido como objetivo principal la comodidad del consumidor, los gustos que posee y la eficiencia necesaria que va simultáneamente con el problema de tiempo del usuario, uno de los contratiempos que tiene como finalidad solucionar la aplicación.

Mediante el transcurso del proyecto se han presentado dificultades principalmente con algunos TDA implementados para la ejecución del programa. Por otro lado se han tenido problemas del tiempo necesario dedicado al desarrollo del programa, debido al tiempo que exigen otras disciplinas y al no tener la noción de cuánto tiempo era necesario dedicar a cada función para que cumplieran con su objetivo.

Una fortaleza notable es la distribución de tareas y funciones dentro del grupo, donde cada integrante tenía claro cual era su trabajo a realizar y cuando tenía que estar terminado. Gracias a esta planificación, cada participante fue capaz de llevar a cabo su labor sin tener que preocuparse por otras partes del programa.

Finalmente, creemos que a futuro se podría mejorar la aplicación ocupando una interfaz más sofisticada, con el propósito de ser más llamativa y fácil de ocupar para el usuario. Esta interfaz nos permitiría implementar las funcionalidades descritas con mayor detalle, ocupando imágenes y gráficos.

## 7. Coevaluación

Mediante la realización del proyecto consideramos que todos los integrantes participaron de forma equitativa en el desarrollo de este, siendo responsables e íntegros con sus tareas correspondientes. Debido a esto, consideramos que no es necesario el sistema de castigo y bonificación ideado.

- **Rodrigo Araos:** Encargado de funciones de añadir películas a un usuario y mostrar el perfil el de un usuario. Responsable con la entrega de material y buen manejo de las Estructuras de Datos. Tiempo extensivo en creación de funciones.

Puntos:  $\pm 0$ .

- **Camila Diaz:** Encargada de funciones de recomendar películas a usuarios según afinidad y mostrar usuarios afines. Responsable con la entrega de material y es activa dentro del grupo para consultas y coordinación. Distribución poco óptima de tiempos para realización de funciones.

Puntos:  $\pm 0$ .

- **Vicente Mercado:** Encargado de funciones de buscar película por ID y mostrar películas de un usuario según criterio. Activo dentro del grupo para consultas y coordinación. Distribución poco óptima de tiempos para realización de funciones.

Puntos:  $\pm 0$ .