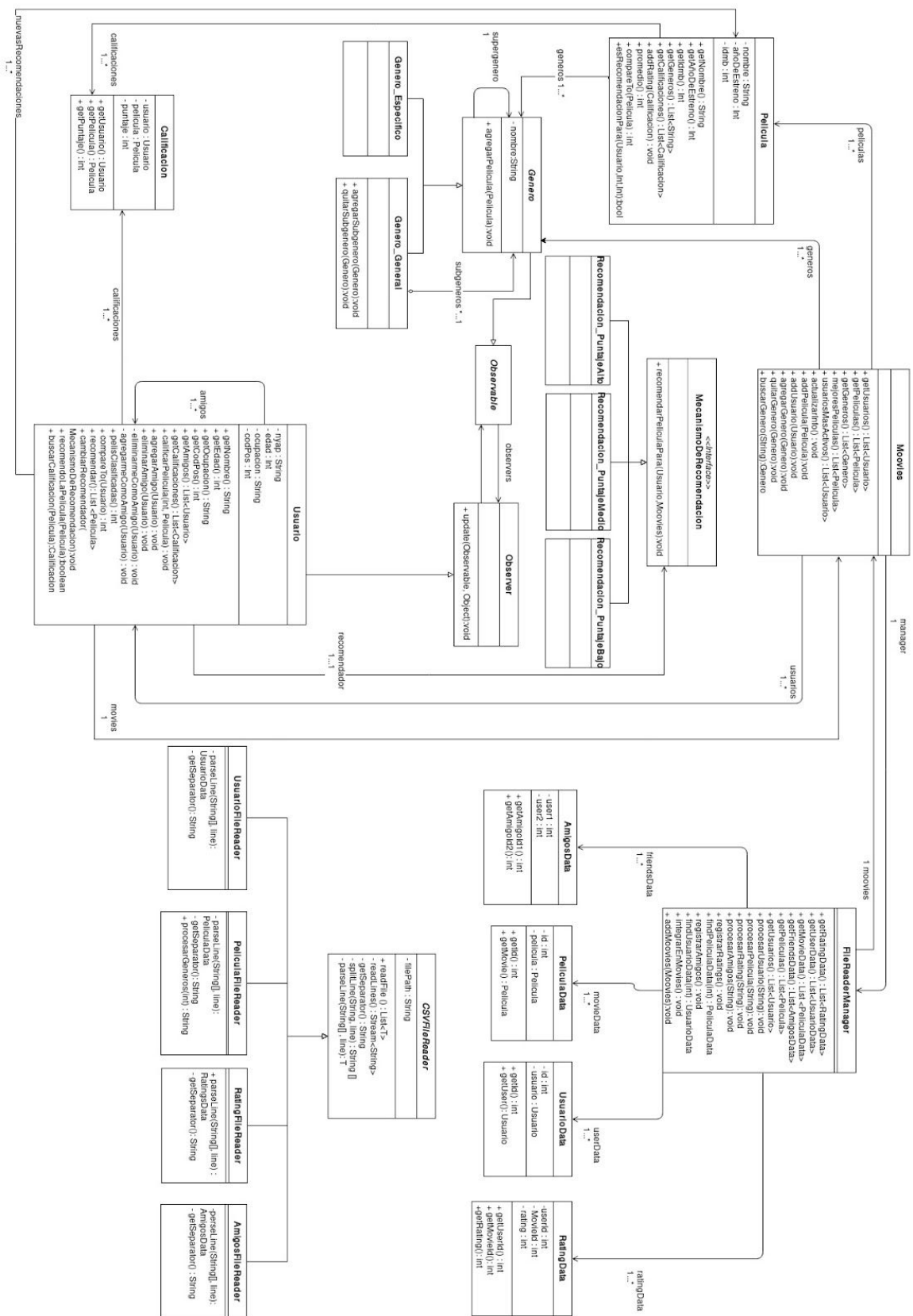


# Hito Final

## Programación con Objetos 2

Tomás Hurrell  
Brian Goldman

## Diagrama UML general:



## **Descripción general del sistema Moovies:**

El objeto principal de nuestro sistema es Moovies. Ésta almacena una lista para de todos los usuarios, otra para todas películas y una mas para todos los generos que hay en el sistema.

Cada Pelicula contiene una lista con todos los géneros que están vinculados a la misma y tiene una lista con todas las calificaciones que recibió.

Los Usuarios por otra parte tienen una lista con todos sus amigos dentro del sistema, y una lista con todas las calificaciones que le dieron a cada una de las películas que puntuaron.

Dichas Calificaciones se encuentran en una clase Calificacion propia, el cual relaciona a la película puntuada con el usuario que calificó la película, y el puntaje que le dio al a misma.

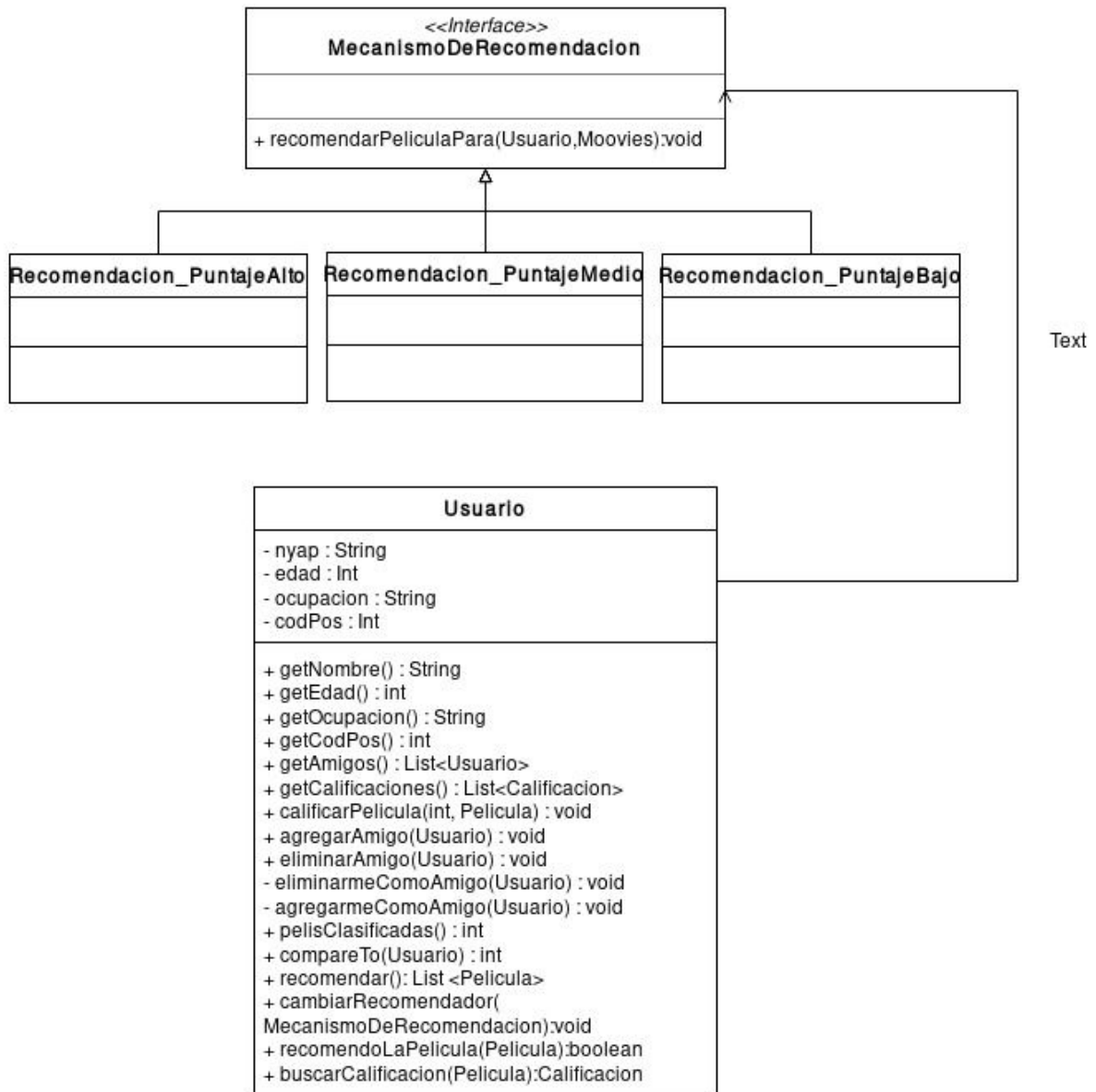
La lectura de los archivos se mantuvo prácticamente igual a los Hitos anteriores: Moovies tiene un objeto FileReaderManager que se ocupa de recolectar la información a importar, en los archivos proveídos. Para esto, crea un objeto específico para cada tipo de archivo que va a leer: PeliculaFileReader, UsuarioFileReader, RatingFileReader, AmigosFileReader. (Los cuatro subclase de CSVFileReader).

Para guardar la información recolectada de dichos archivos, el FileReaderManager guarda cuatro listas de clases encargadas de almacenar y asociar la información: UsuariosData, PeliculasData, RatingData, AmigosData.

## **Recomendaciones:**

Para las recomendaciones de películas, creamos una interfaz MecanismoDeRecomendacion, que define el mensaje recomendarPeliculaPara(Usuario, Moovies). Además, hay tres clases que implementan las diferentes estrategias de recomendación, implementando el método de este mensaje: MecanismoDeRecomendacion\_PuntajeBajo, MecanismoDeRecomendacion\_PuntajeMedio y MecanismoDeRecomendacion\_PuntajeAlto.

Esto respeta el patrón de diseño Strategy, ya que las diferentes estrategias de recomendación de películas, quedan cada una, aislada en una clase diferente, respetando el polimorfismo de la interfaz MecanismoDeRecomendacion.

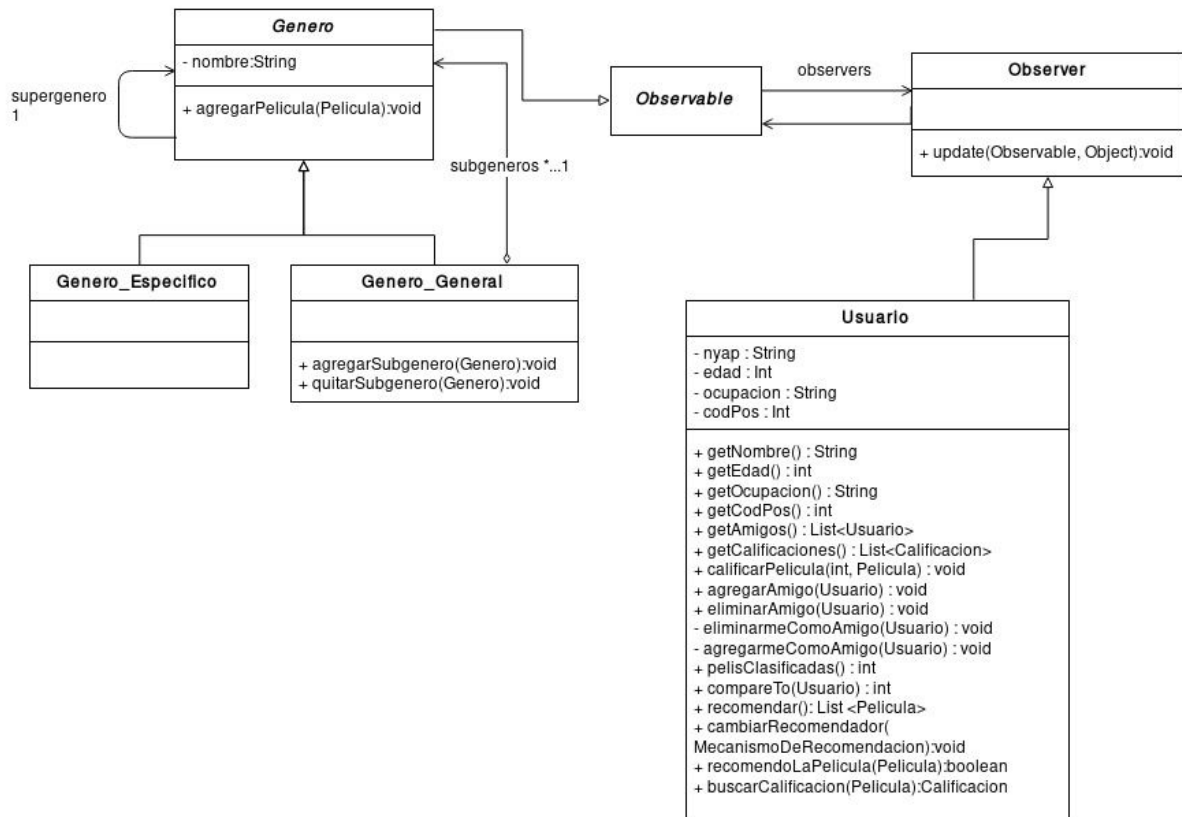


En este caso, podemos ver que la clase **Usuario** sería nuestro Client, **MecanismoDeRecomendacion** juega el papel de Abstraction, y las tres estrategias de recomendación (cabe aclarar que se pueden agregar otras clases que respeten la interfaz, cada una con su propia estrategia de recomendación específica) juegan el papel de las Implementations o estrategias concretas.

Además el **Usuario** puede intercambiar entre los mecanismos de recomendación utilizando el mensaje `cambiarRecomendador(MecanismoDeRecomendacion)`

## Generos:

Respecto al diseño de los Generos, utilizamos el patrón Composite, y respecto a la suscripción de Usuarios a los distintos géneros disponibles, utilizamos el patrón Observer.



Para los generos tenemos una clase abstracta Genero, que define el mensaje común a todos los géneros `agregarPelicula(Pelicula)`, que se utiliza cada vez que se introduce una nueva Pelicula en el sistema de Moovies. Además tenemos dos subclases de genero, **Genero\_General**, que representa a todos aquellos géneros que tienen subgéneros, (por ejemplo Terror) y **Genero\_Especifico** que representa a aquellos géneros que no tiene subgéneros, (por ejemplo Vampiros).

A su vez todo Genero tiene un colaborador interno **supergenero**, que es el género que lo abarca. Cuando el género no es abarcado por nadie, se tiene a sí mismo, en dicho colaborador interno.

En este caso, Genero hace las veces de la interfaz Component, Genero\_Especifico hace las veces de Leaf, y Genero\_General hace las veces de Composite respetando la recursividad de dicha estructura.

Por otro lado, respecto de las suscripciones de películas, los Usuarios son subclase de Observer, es decir, son observadores específicos de los Generos, mientras que la clase

Genero es subclase de Observable, lo que permite que ante cualquier cambio en los Generos al agregar una nueva película, se notifique a todo observador (los Usuarios) y estos, gracias al método update hagan lo que necesiten, en este caso particular, agregar la película a la lista de nuevasRecomendaciones que tiene cada Usuario.