**Base de datos 1**

Alumno: Santiago Vietto

Docente: Leandro Luis Juárez

DNI: 42654882

Institución: UCC

Año: 2021

**Ejercicio 1: Gestión de un Video Club**

**Parte 1**

\_ Para el desarrollo de un sistema para gestionar un video club se pide diseñar la base de datos del mismo, cumpliendo los siguientes requerimientos. Para las **películas** se pide almacenar:

* Código de la película (dato interno del video club)
* Nombre de la película
* Director
* Actores (Tiene varios)
* Fecha de estreno en el cine
* Valoración del público. “estrellas” ( valor del 1 al 5)
* Género

*- (Asumimos que tenemos 1 solo ejemplar de cada película)*

*- (Solo es necesario guardar los 3 actores más importantes)*

\_ Para los **clientes** del video club se pide:

* Nombre
* Apellido
* Dirección
* DNI
* Teléfono
* Email
* Genero preferido

\_ Para los **usuarios** del sistema:

* Nombre
* Apellido
* Contraseña
* Turno
* Dirección
* Nivel de acceso (Completo, Parcial, Limitado)

\_ Para el registro de **alquileres** es necesario registrar que película se alquiló, quien lo hiso y que usuario fue el responsable del alquiler. También se debe registrar la fecha del alquiler, fecha de devolución y el estado del mismo (En espera de devolución, Demorado, Entregado).

* + *(Se asume que cada alquiler tiene solo una película)*

**Parte 2**

\_ Cuáles son las modificaciones necesarias que tendríamos que realizar en el diagrama E-R previamente diseñado para que:

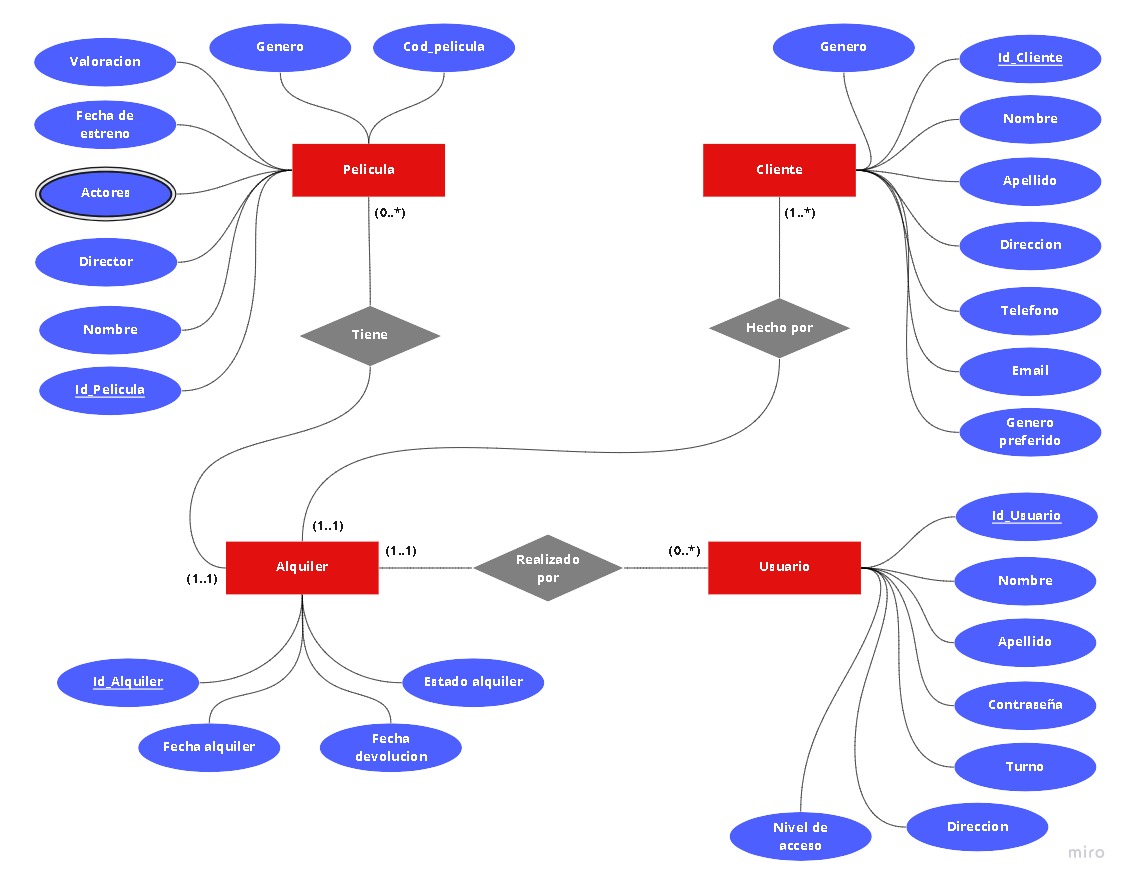
* Los alquileres contemplen más de una película.
* Los clientes, al hacer una devolución, pueden devolver todas las películas o solo algunas.

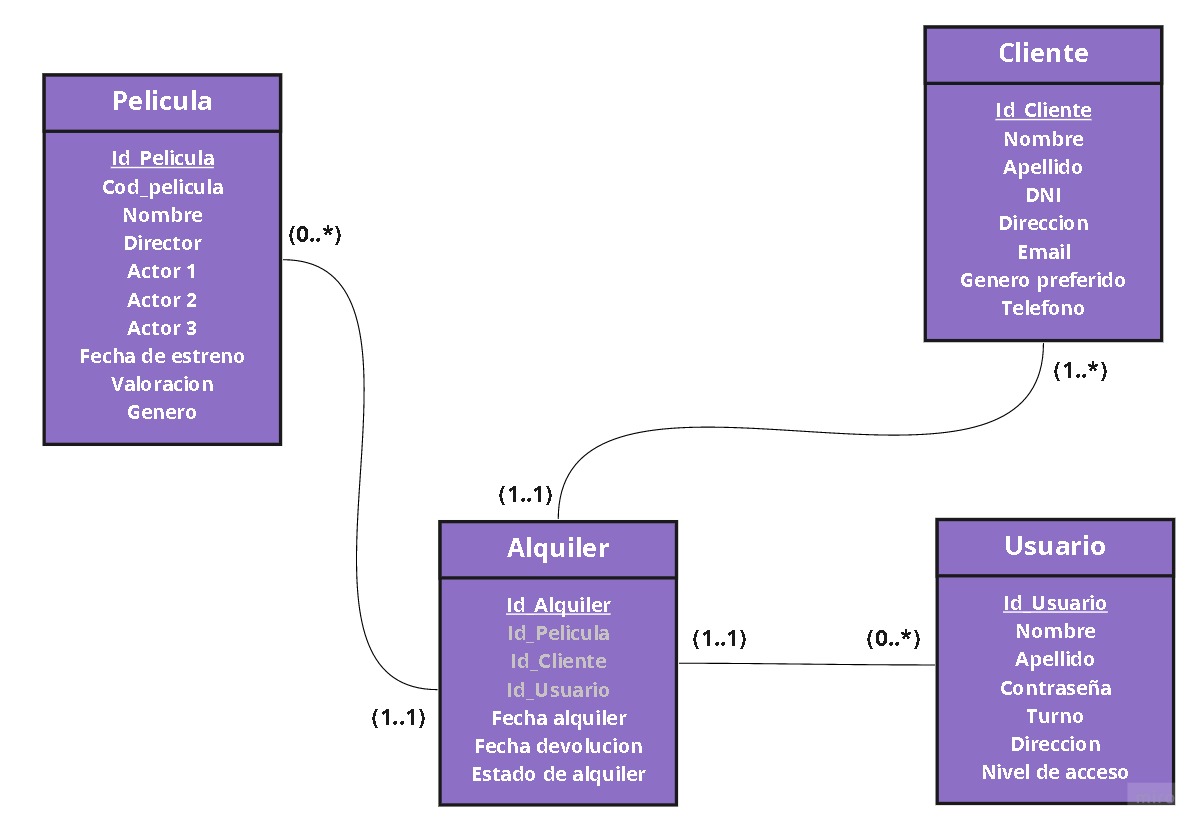
**Resolución parte 1**

\_ En la entidad película no usamos el código de película como Id porque nosotros no sabemos si efectivamente el código de la película es univoco para identificar a cada película porque capaz que en el video club se usa el mismo código para dos películas distintas pero que están en distintos estantes, entonces para evitar esas cosas tenemos nuestro propio Id. El código de película es un dato que el usuario maneja, o sea que cuando entra una película nueva él le pone el título que quiera, por eso no nos fiamos de esos datos como desarrolladores.

\_ En la entidad usuario, el atributo nivel se acceso no es compuesto, porque este es un atributo que dentro lleva un dato, es decir, los tres datos que mencionan son los valores que toma el atributo. Esto es similar a la valoración en la entidad película. El cliente no sabe lo que es un atributo, por eso si nos pide que tres usuarios van a tener acceso, a nosotros no nos importan quienes son, simplemente creamos la entidad usuarios, y despues el cómo se llaman es un dato que va dentro del atributo nombre. Entonces asi como no me importan los nombres de los usuarios, no me interesan los niveles de acceso que tenga, ya que simplemente son el valor que toma el atributo.

\_ Con respecto a las restricciones cada alquiler tiene una y solo una película. Que las películas tengan (0..\*), que es la relación entre la película y el alquiler, significa que tenemos una lista de películas y tenemos películas que nunca se alquilaron, de ahí el 0, y él muchos \* significa que una película se pudo alquilar muchas veces. Si en vez de un 0 hay un 1 no significa que todas fueron alquiladas, sino que toda mi lista de películas alguna vez una se alquiló, al menos una vez. En el cliente, el 1 significa que no está cargado en el sistema clientes que nunca alquilaron, entonces todos los clientes que están cargados en el sistema al menos una vez han hecho un alquiler o varios, y por otro lado un alquiler tiene uno y solo un cliente. El alquiler también tiene uno y solo un usuario, y en los usuarios, el (0..\*) significa que se dio de alta varios empleados o ninguno.

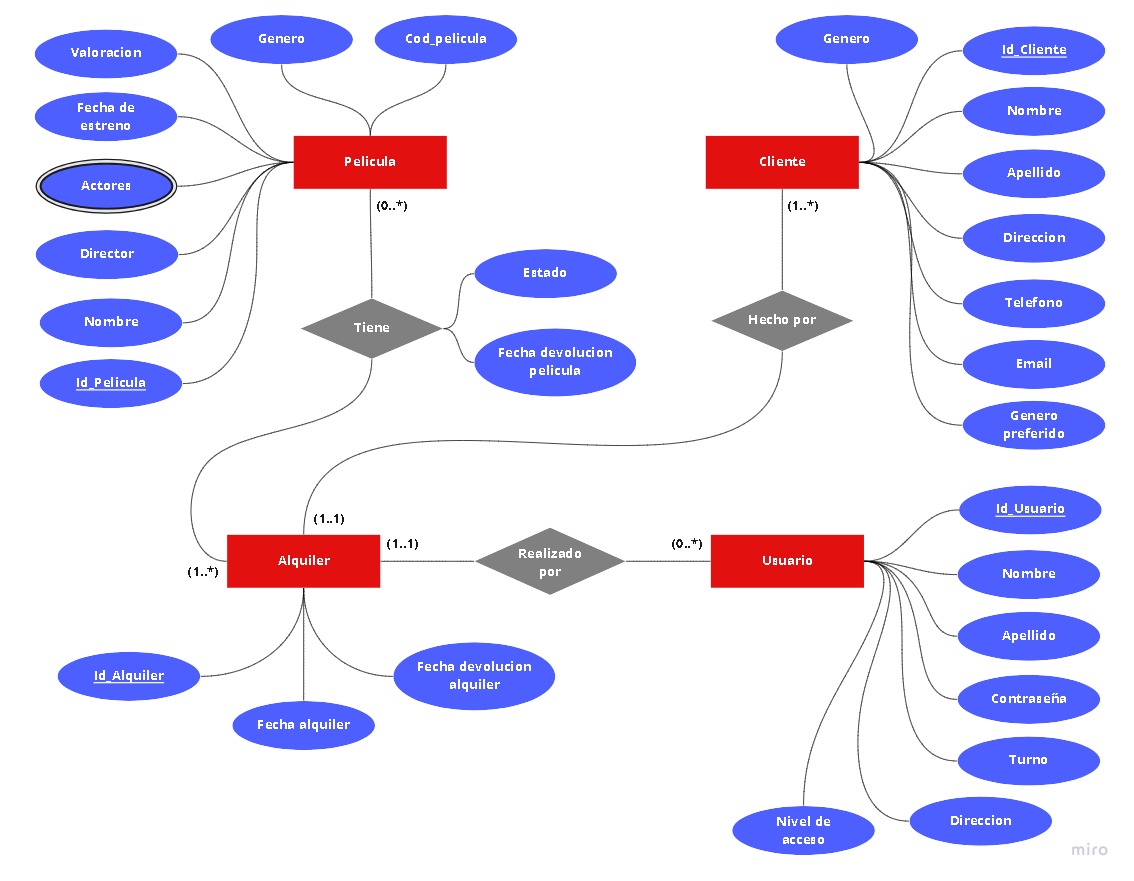
\_ Por ende el diagrama E-R quedaría:

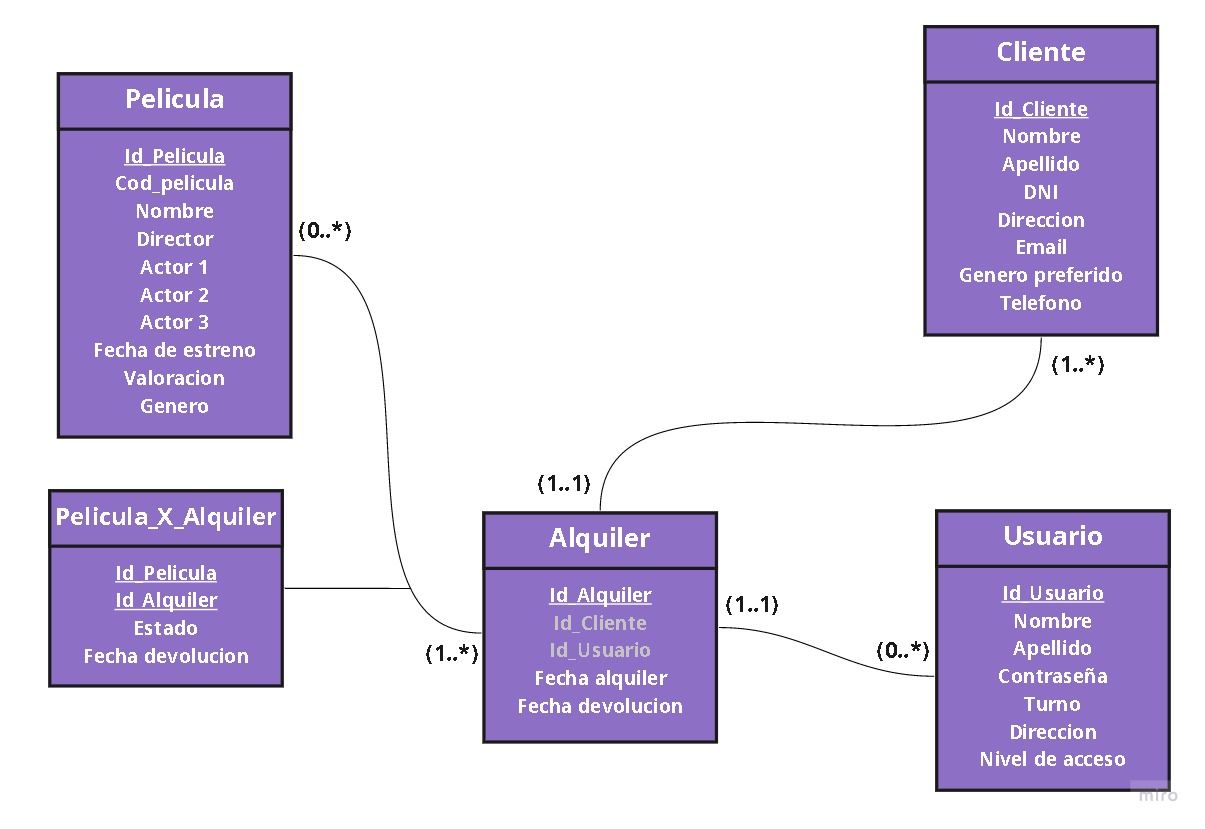
\_ El diagrama de clases quedaría:

**Resolución parte 2**

\_ Si en la entidad alquiler ponemos las películas alquiladas (Id\_Pelicula 1, Id\_Pelicula 2, etc) más las fechas de devoluciones de cada una, la tabla se vuelve muy grande y no es una solución performante. La fecha de devolución ya no nos sirve, porque ahora al poder alquilar varias películas y que se puedan devolver a distintos tiempos, esa fecha de devolución nos quedaba de alguna forma obsoleta. Lo mismo con el estado del alquiler, ya que cada película al tener una devolución distinta podemos tener diferentes estados, entonces este atributo también queda obsoleto.

\_ El problema lo tenemos entre la entidad película y la entidad alquiler. Y acá es donde aparecen atributos, que son de la relación entre ambos. La fecha de devolución no puede ser de la película porque esta se puede alquilar un montón de veces y del alquiler tampoco puede ser porque este puede tener una fecha de devolución fija pero despues cada elemento de la relación tiene su propia fecha de devolución. Es por esto que a la fecha devolución la ponemos en la relación, en donde además agregamos un atributo estado, que es el estado de una película en su alquiler, y lo mismo seria con la fecha de devolución de una película en su alquiler. Y por otro lado le agregamos a la entidad alquiler una fecha de devolución propuesta o pactada del alquiler entero, a diferencia de la otra fecha que es de una película individual, que pueden coincidir, por ejemplo alquilo 10 películas y las devolvió de una a todas, o devolvió algunas y se demoró en entregar otras, etc.

\_ El diagrama E-R seria entonces:

\_ El diagrama de clases seria: