



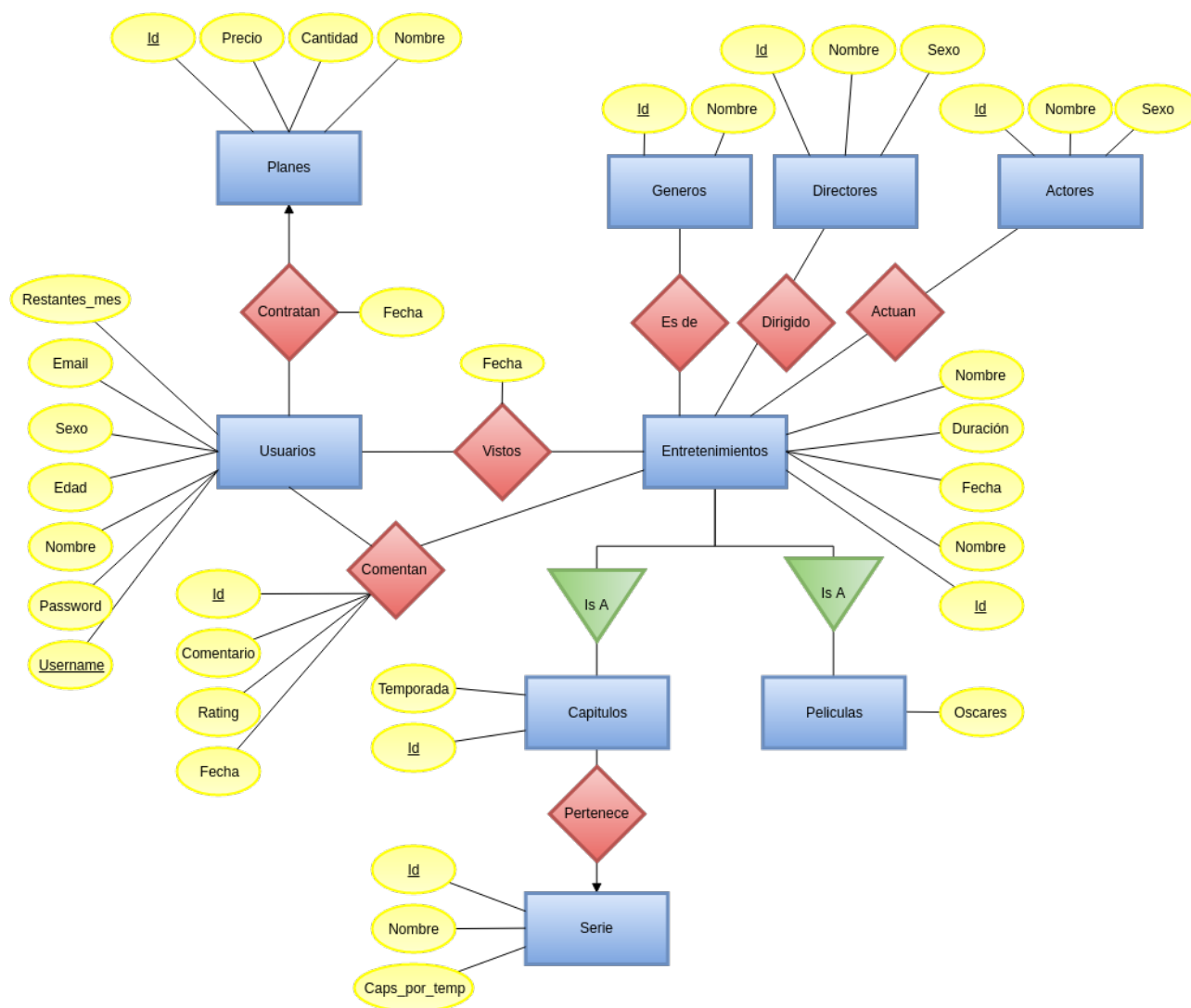
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2413 —Bases de Datos — 1' 2016

Entrega 2

Se escogió la opción **2** para el desarrollo de esta entrega.

El diagrama entidad-relación es el siguiente:



En el diagrama, manteniendo similitud con la notación vista en clases, se presentan de color azul las entidades, los atributos de amarillo y las relaciones en rojo. Las flechas indican la cardinalidad de las relaciones, por lo que una flecha apuntando a una entidad, implica que la entidad apuntada es de cardinalidad 1 y la que apunta N . Por otro lado aquellas uniones o conexiones sin flecha indican relaciones N a N . No hay en este caso relaciones 1 a 1.

Las relaciones presentadas en la mayoría de los casos no llevan señalados sus atributos. Más adelante se constata que sí tienen atributos, pero estos corresponden a las llaves de las entidades que relacionan. En los casos que la relación agrega información, como es el caso de la entidad *Comentan* con el atributo *Fecha*, se señala en el diagrama.

La estrategia general usada para fabricar el diagrama fue pensando en eliminar redundancias y generar a futuro tablas que no tengan más información de la necesaria, de cierto modo, ceñirse a su nombre, es decir, si una tabla es de *Usuarios*, que no tenga atributos propios de otras entidades, tal como el *precio* que paga por el *Plan* contratado.

A continuación se presenta una tabla con la descripción de las tablas incluidas en el diagrama:

Tabla	Descripción
Usuarios	La tabla contendrá a todos los registrados en la página, con los datos para acceder, información general y un atributo de <i>restantes_mes</i> , correspondiente a la cantidad de <i>capitulos</i> y/o <i>series</i> que puede ver aún, considerando la regla de 1:4 (<i>Pelicula</i> : <i>Capitulo</i>)
Planes	Esta tabla presenta la información de los planes de <i>MaquiView</i> , donde el atributo cantidad representa el número de <i>capitulos</i> y/o <i>series</i> con los que cuenta un usuario al contratar un plan
Entretenimiento	Muestra la información genérica a un capítulo o a una película, información referida al contenido audiovisual, duración, fecha, nombre, etc.
Capitulos	Contiene, además de los atributos comunes a cualquier contenido audiovisual (entretenimiento), el número de la temporada a la que pertenece.
Series	La tabla presenta el nombre de la serie, junto con la cantidad de capitulos por temporada.
Peliculas	Presenta la información básica de contenido multimedia, pues hereda de entretenimiento y además un atributo oscars para saber la cantidad de premios con que cuenta.
Actores	Muestra el nombre de los actores y su sexo, para así poder generar consultas relacionadas con el género de los actores/actrices.
Directores	Nuevamente, el nombre de los directores, junto con su sexo, por la misma razón.
Generos	Se incluye como tabla, ya que en el enunciado se señalaba la posibilidad de pertenecer a varios géneros.

Todas las tablas anteriores cuentan con un identificador (*id*) que es la llave de ellas, excepto la tabla *Usuarios*, cuya llave es el *username*.

Adicionalmente las tablas generadas por las relaciones son:

Tabla	Descripción
Comenta	Usuario comenta entretenimiento, tiene el <i>username</i> del usuario, <i>id</i> del entretenimiento, <i>id</i> del comentario, comentario en sí, y rating, más un atributo adicional de la fecha en que se realizó.
Contrata	Usuario contrata un plan, tiene el <i>id</i> del plan y <i>username</i> del usuario, con fecha, no pensando en fechas de expiración ya que los planes terminan por un tema de cantidad de consumo, sino que pensando en las estadísticas (usuarios suscritos a alguna fecha por ejemplo).
Vistos	Usuario ve entretenimiento (capítulo/película), tiene el <i>id</i> de ambos.
Dirige	Actor dirige entretenimiento, con el <i>id</i> de ambos.
Es de	Entretenimiento es de un género, tiene <i>id</i> de ambos.
Actuan	Actor actúa en entretenimiento, con el <i>id</i> de ambos.
Pertenece	Capítulo pertenece a serie, nuevamente, tiene el <i>id</i> de ambos.

En las tablas correspondientes a relaciones se trabaja con llaves foráneas cuando se menciona "el *id* de ambos" o similar.

Las dependencias funcionales son las siguientes, se presentan algunas, ya que las demás son análogas, la intención es mostrar que las llaves determinan los otros atributos:

Tabla	Dependencia
Usuario	$username \rightarrow \text{nombre, sexo, correo, password, edad, email}$
Planes	$id \rightarrow \text{nombre, precio, cantidad}$
Entretenimientos	$id \rightarrow \text{nombre, duración, fecha}$
...	

Y el modelo se encuentra en BCNF porque, las dependencias funcionales presentan llaves a la izquierda de la relación, y la forma en la que se armaron las tablas fue generando tablas para las relaciones, lo que nos asegura no agregar redundancia ni dependencias que violen BCNF.

Los comandos utilizados fueron en general muy similares entre grupos, se incluyen separados por grupos de características similares:

1. Tablas con primary keys

```
CREATE TABLE Usuarios(username varchar(15) PRIMARY KEY, password varchar(15), nombre varchar(40), sexo char(1), edad int, email varchar(50), restantes_mes float);
CREATE TABLE Planes(id int PRIMARY KEY, nombre varchar(40), precio float, cantidad int);
CREATE TABLE Entretenimientos(id int PRIMARY KEY, nombre varchar(40), duracion int, fecha date);
CREATE TABLE Generos(id int PRIMARY KEY, nombre varchar(40));
CREATE TABLE Actores(id int PRIMARY KEY, nombre varchar(40), sexo char(1));
CREATE TABLE Directores(id int PRIMARY KEY, nombre varchar(40), sexo char(1));
CREATE TABLE Series(id int PRIMARY KEY, nombre varchar(40), caps_por_temp int);
```

2. Tablas con herencia (modelada como una FOREIGN KEY)

```
CREATE TABLE Capitulo(id_entretenimiento int PRIMARY KEY REFERENCES Entretenimientos(id), temporada int);
CREATE TABLE Peliculas(id_entretenimiento int PRIMARY KEY REFERENCES Entretenimientos(id), oscars int);
```

3. Tablas con llaves foráneas

```
CREATE TABLE Contrata(id_plan int REFERENCES Planes(id), username varchar(15) REFERENCES
    Usuarios(username), fecha date);
CREATE TABLE Comenta(id int PRIMARY KEY, username varchar(15) REFERENCES Usuarios(username),
    id_entretenimiento int REFERENCES Entretenimientos(id), comentario text, rating float,
    fecha date);
CREATE TABLE Vistos(id_entretenimiento int REFERENCES Entretenimientos(id), username varchar
    (15) REFERENCES Usuarios(username), fecha date);
CREATE TABLE Dirige(id_entretenimiento int REFERENCES Entretenimientos(id), id_director int
    REFERENCES Directores(id));
CREATE TABLE EsDe(id_entretenimiento int REFERENCES Entretenimientos(id), id_genero int
    REFERENCES Generos(id));
CREATE TABLE Actuan(id_entretenimiento int REFERENCES Entretenimientos(id), id_actor int
    REFERENCES Actores(id));
CREATE TABLE Pertenece(id_capitulo int REFERENCES Capitulo(id), id_serie int REFERENCES Series
    (id));
```

En cuanto a alguna consulta de las iniciales correspondiente a lo definido para la entrega 1, tenemos que, por ejemplo, la consulta referida a, dado un usuario obtener su plan y sus restantes sería:

```
SELECT planes.nombre, usuarios.restantes_mes FROM usuarios, contrata, planes WHERE contrata.
    username='pepe' and contrata.username = usuarios.username and planes.id = contrata.id_plan;
```