



Práctica Intermedia

Diagrama modular

Ambiente: Cine

Entradas:

Cliente
 |->Su informacion
 |->Usuario
 |->Contraseña
 |->Informacion de Accion
 |->Clicks | Dispositivos de
 |->Teclado | entrada
 |->Pantalla Tactil.

Salas	
	Nombre/Identificador
	No. total de asientos
	Estado (Mantenimiento/Funcionando)

Asientos	
	Asiento_ID/Numero de asiento
	Estado (Disponible/No Disponible)

Horarios	
	Sala_Id
	Pelicula_ID
	Fecha
	Hora

Salidas:

Cliente
 |->Boleto
 |->Titulo de pelicula
 |->Numero de asiento
 |->Sala
 |->Horario
 |->Fecha

Boletos	
	No. Boleto/Boleto_ID
	No. Usuario
	Pelicula
	Sala
	Asiento
	Fecha
	Hora

Usuarios	
	ID_Usuario
	Nombre
	Correo
	Telefono
	Contraseña
	Tipo de Usuario

Peliculas	
	Pelicula_id
	Titulo
	Duracion
	Clasificacion
	Sipnosis
	Formato
	Idioma
	Fecha Lanzamiento
	Director/es
	Actores

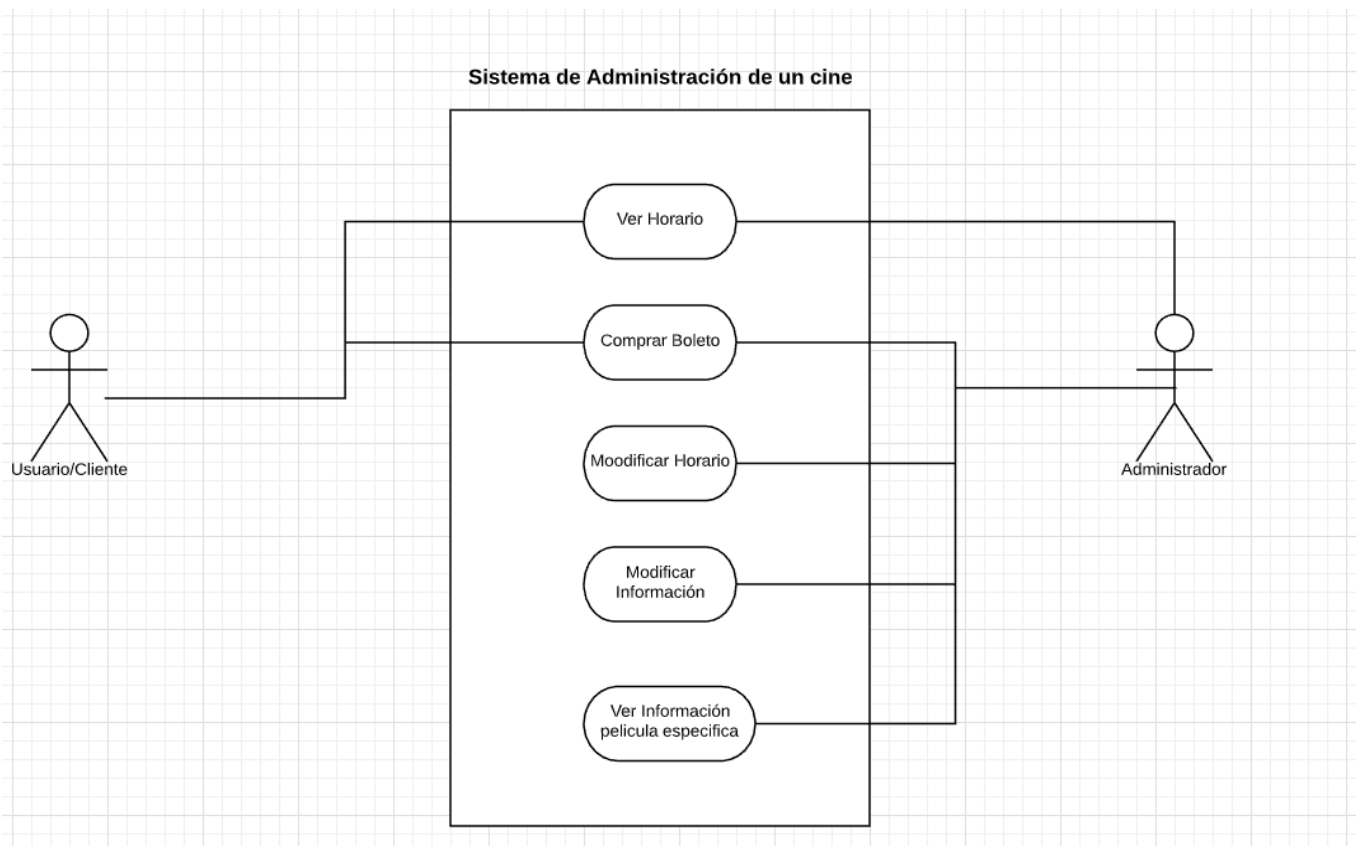
Actores:

Usuarios General(Cliente), Administradores

Lo primero que tenemos es el diseño del sistema en un diagrama modular, que tiene cuáles serían las entradas y salidas, así como las posibles tablas que se requieren para realizar el módulo de la compra de un boleto. De manera resumida, lo que se busca lograr con el módulo es que el usuario/cliente pueda checar los horarios de las películas y comprar un boleto si lo desea.



Diagrama Casos de Uso



Después tenemos nuestro diagrama de caso de uso, en el cual solo se agregan los casos que incluyen las acciones generales del administrador a lo relacionado con los horarios de películas o información que se relaciona con éstas. Para el usuario solo se toma en cuenta el poder ver los horarios de las películas que se proyectan actualmente y el poder comprar un boleto, el cual será el caso que se desarrollará y con el que se trabajará en la implementación en código en esta práctica.



Casos de Uso

Nombre: Comprar Boleto

Suposiciones:

- Se ha ingresado a la opción de comprar boleto.
- Se ha iniciado sesión.

Precondiciones:

- Se han checado los horarios de las películas.
- Hay asientos disponibles para esa película.

Inicio:

- Se da click en la opción de comprar boleto

Flujo de eventos:

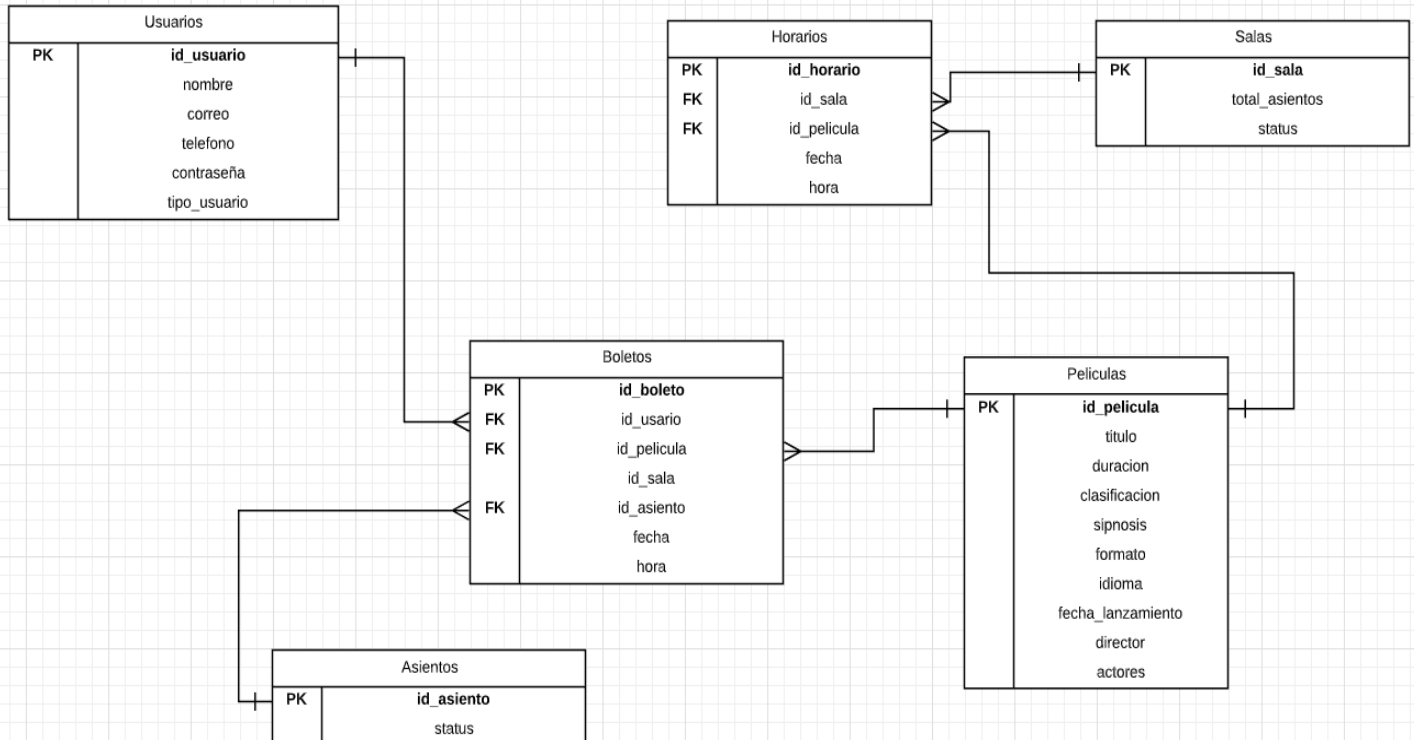
- **S:** Despliega una lista de los títulos de películas disponibles.
- **U:** Selecciona la película que desea ver.
- **S:** Despliega una lista de horarios disponibles para esa película.
- **U:** Selecciona el horario de su preferencia.
- **S:** Despliega una lista de asientos disponibles.
- **U:** Selecciona el asiento que desee.
- **S:** Muestra información seleccionada
- **S:** Pide confirmación al usuario finalizar su compra
- **U:** Verifica información.
- **U:** Confirma su compra.
- **S:** Genera boleto para el usuario.

Postcondición:

- Se actualiza la base de datos agregando los datos de un nuevo boleto vendido y actualizando las tablas que se relacionen a este, como la de asientos cambiando su estado de disponibilidad.



Diagrama Relacional



Primeramente, tenemos el diagrama principal sin normalizaciones, solo con las relaciones las cuales son las siguientes:

Usuarios-Boletos: Se uso una relación 1 a muchos ya que un usuario puede adquirir varios boletos.

Asientos-Boletos: Al inicio se pretendía usar una relación 1 a 1, ya que un asiento solo puede estar en un boleto, pero esto solo funcionaria si hablamos de una sola función, ya que al transcurso del día el asiento se desocuparía y puede aparecer en otro boleto en una distinta función, por eso se decidió al final dejarlo como 1 a muchos.

Salas-Horarios: Se uso una relación 1 a muchos ya que una sala puede estar en varios horarios diferentes en el transcurso del día.



Películas-Horarios: Al igual que la relación anterior, es 1 a muchos ya que una película puede estar en diferentes horarios en el día.

Películas-Boletos: Finalmente, esta relación es 1 a muchos ya que una película puede aparecer en distintos boletos.

Normalización de las tablas

1NF:

El siguiente paso es normalizar las tablas, primero aplicaremos la 1NF.

Usuarios	
PK	id_usuario
	nombre
	apellidoP
	apellidoM
	correo
	telefono
	contraseña
	tipo_usuario

El único cambio sería en la tabla usuarios, en la cual el nombre se puede separar para nuestra región en los apellidos. Las otras tablas, aunque algunos atributos se podían cambiar, como la fecha, se decidió dejar así por conveniencia, ya que al generar la base de datos contamos con un tipo de dato que nos permite manipular la fecha completa y no necesariamente tener que separarla en día/mes/año. Otro punto que se mencionará es en la tabla de películas, que, si bien el atributo actores y director se puede separar en mas partes, en esta ocasión se manejará como un atributo de tipo “descripción” que solo será un texto con los nombres de las personas correspondientes, sin necesidad de brindar algún tipo de información extra para tener que dividirlo o agregar más datos.



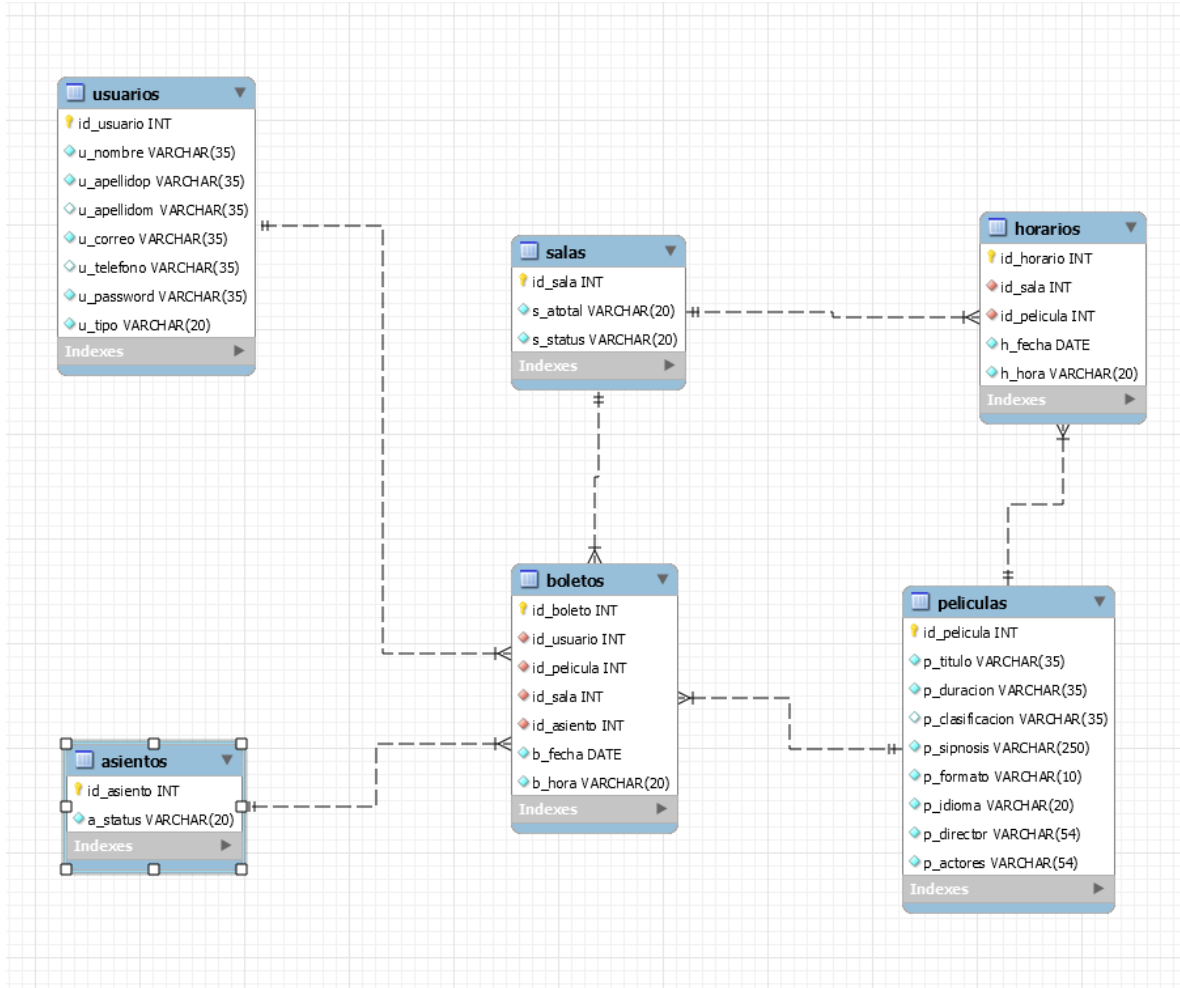
2NF:

Para realizar la 2NF revisamos que se cumpla en todas las tablas la condición correspondiente. En todas se cumple, pero había ciertas dudas de si usar una llave compuesta en lugar de un identificador para los horarios, pero finalmente se optó por un identificador para manipular más fácilmente los datos ya que ningún identificador se puede repetir por el cambio sujeto a duración de películas, la fecha o incluso si se mueve la sala, difícilmente un horario sería igual. De igual manera con la tabla de películas, se pudo usar como llave primaria el título, ya que son ese dato se pueden identificar los demás y no se puede repetir, pero al ser una cadena de caracteres y no siempre de la misma longitud puede dificultarnos la manipulación de los datos con este tipo, ya que también es usada en dos relaciones diferentes y al final se optó por usar un identificador como algún número o tipo de “folio”. De este modo queda normalizada en 2NF.

3NF:

Finalmente, la 3NF ya no se aplicó, ya que, aunque la mayoría de las tablas se cumple la condición extra, en la tabla “películas” no se cumple por tener el atributo título además del identificador de la película como PK, ya que con este se pueden conocer todos los demás datos y hacer una tabla extra solo para separar el título se considero algo innecesario. Para no crear tablas de más, se optó por usar la opción de dejarla hasta 2NF.

En lo que respecta a los diagramas eso sería todo el diseño, lo siguiente es pasar a la implementación del código.



Finalmente, este es el diagrama que obtuvimos en la base de datos final.

Nota: Se hicieron algunas modificaciones al diseño inicial para poder manipular los datos y para corregir algunos errores con el código.