

# Sistema de información MVC para el manejo y control de un cine.

Marco Jair Rodríguez Hernández.  
(146012)  
*Lic. Ingeniería en Sistemas Computacionales*  
Universidad de Guanajuato.

## Sistemas de Información.

### Introducción.

Un sistema de información es un conjunto de componentes interconectados que colecciona, recupera, distribuye, almacena y procesa la información que se obtiene a partir de una entrada.

Modelo Vista Controlador (MVC) se refiere a un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El *Modelo* contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

La *Vista*, o interfaz de usuario, compone la información que se envía al cliente y los mecanismos interacción con éste.

El *Controlador*, actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

En conjunción, un sistema de información MVC, es un sistema capaz de procesar la información proveniente de una interfaz de usuario a través de distintas técnicas de lógica de control programadas enviando la información y colectando los datos necesarios para dar una salida.

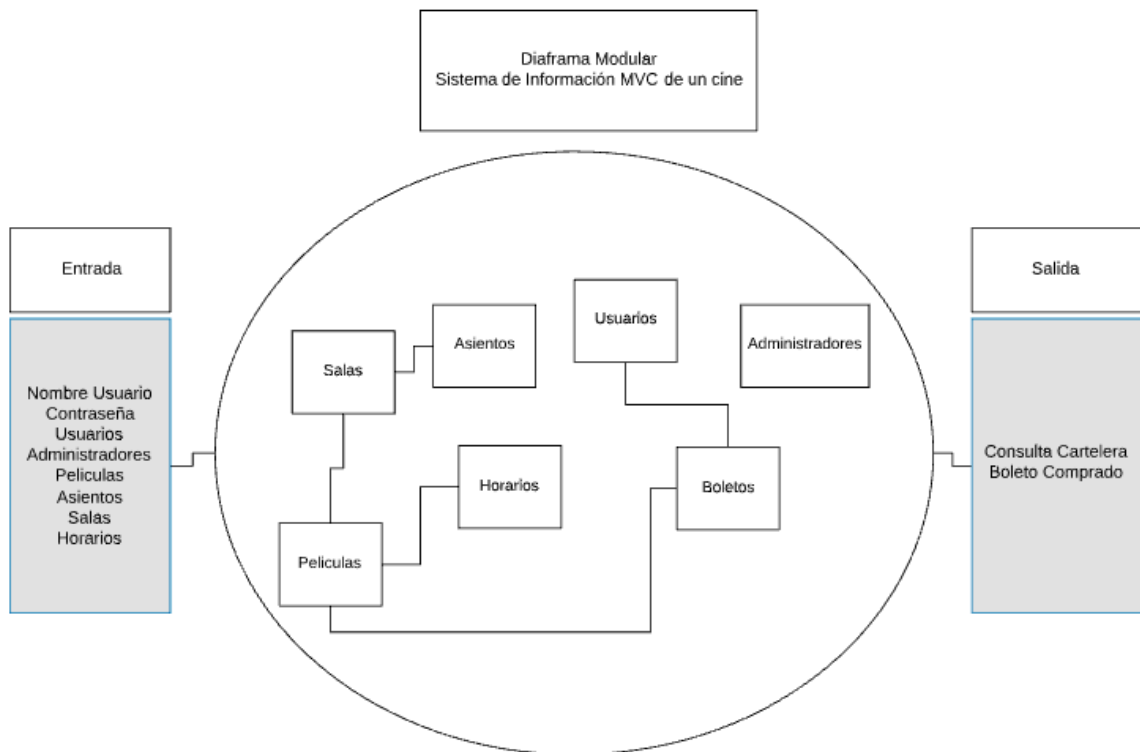
En este caso, hablaremos del comportamiento de un sistema MVC para el manejo y control de un cine. El sistema es capaz de administrar películas, salas, asientos, horarios, usuarios y administrativos que conforman un cine. Cada atributo es un componente clave en el sistema y la forma en que se estructura el mismo.

## Desarrollo.

### Diagrama modular.

El diagrama modular presenta a grandes rasgos la estructura que conforma el sistema de información, cada modulo que se interconecta con otro hace referencia a las posibles relaciones que existen en el sistema.

Con el diagrama modular podemos darnos una idea de cómo se procesará la información en el sistema y ver que tipo de comportamiento va tomando para otorgar al usuario una salida pertinente.



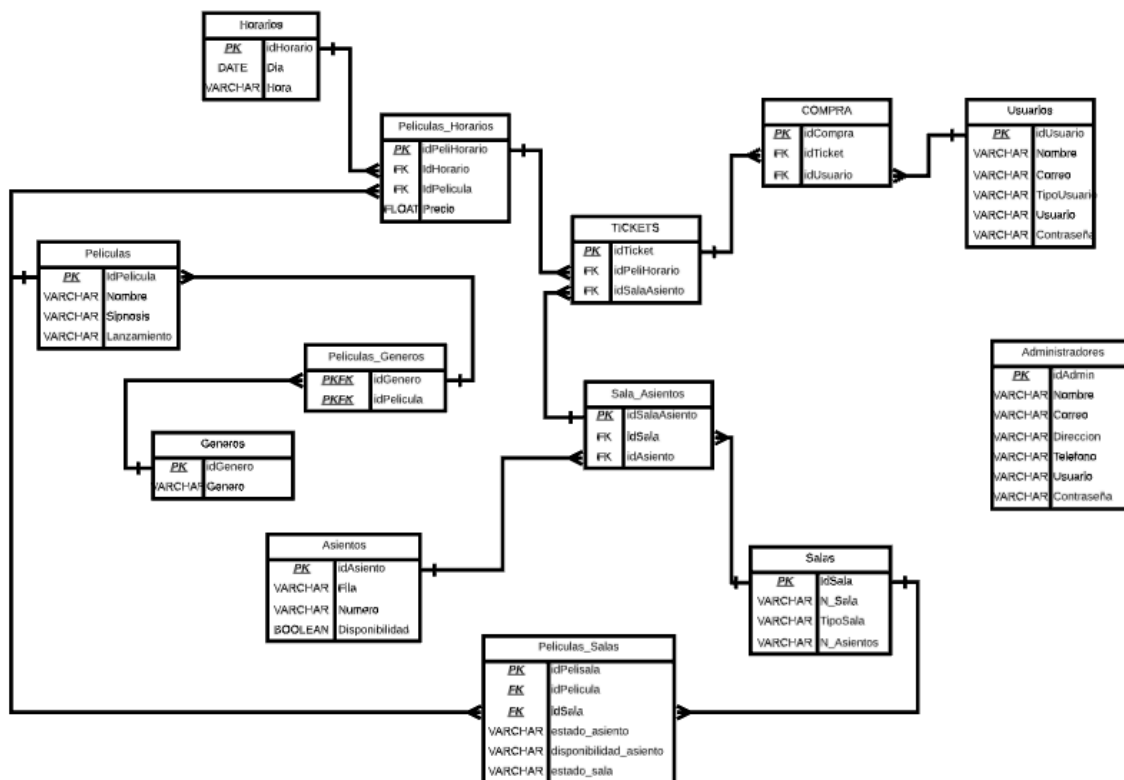
**Figura 1.** *Diseño modular del sistema de información MVC de un cine.*

## Diagrama Relacional.

El diagrama relacional a diferencia del modular, presenta las relaciones que sigue la base de datos indicando las llaves primaria de cada entidad y las llaves foráneas para el manejo de la información.

Gracias a este tipo de diagramas podemos sacar conclusiones de cómo podemos interconectar los atributos de una entidad y usarlos de forma pertinente a nuestro beneficio.

En general, nos permite observar de manera mas detallada el comportamiento de la información en el sistema.



**Figura 2.** Diseño relacional del sistema de información MVC de un cine.

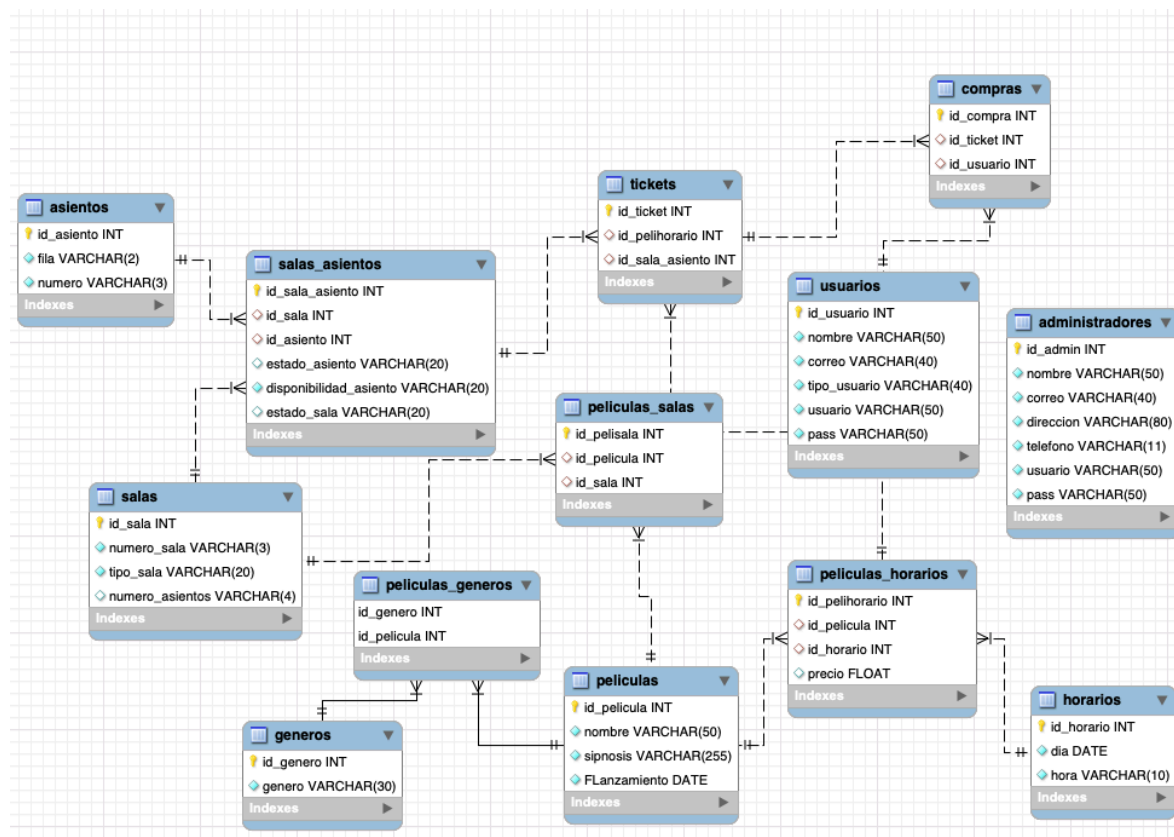
La Fig. 2 muestra el diagrama relacional utilizado para la creación de la base de datos del sistema.

El diagrama relacional no se encuentra normalizado, ya que en las tablas usuarios y administradores no se descompone el campo nombre por “Nombre (S)” y “Apellidos” asimismo el campo dirección en la tabla administradores no se descompone por los campos

“Calle, colonia, CP, Núm. Ext, Núm. Int, etc.” así como en la tabla películas “Lanzamiento” no se descompone por “Día, Mes y Año”.

Se optó por esta estructura ya que se considera que para un cine es irrelevante que el usuario ingrese su nombre completo o simplemente un apellido al igual que en una película ingrese su año de lanzamiento por día, mes, año que una fecha de estructura “AAAA-MM-DD” así como que un administrador registre su dirección por campos “calle, colonia, núm. ext., etc.”. Sin embargo, las demás tablas cumplen con las condiciones de la primera forma normal (1NF) pero como se mencionó desde un principio el diagrama relacional no se encuentra normalizado.

Base de datos.



**Figura 3.** Base de datos del sistema de información MVC de un cine.

La Fig. 3 muestra de forma gráfica el desarrollo final de una base de datos creada desde MySQL. Se puede apreciar que sigue exactamente el mismo comportamiento que el diagrama relacional.

Descripción de las tablas de la base de datos.

El comportamiento de las relaciones en las tablas es fundamental para el manejo de la información tanto del usuario como del sistema.

### **Tabla Usuarios.**

La tabla de usuarios nos permite almacenar la información directa de un usuario, en ella encontramos los siguientes campos.

***Id\_usuario:*** este campo es de tipo entero, único e irrepetible y nos permite tener un registro único del usuario, al ser único e irrepetible es nuestra llave primaria.

***Nombre:*** este campo almacena el nombre completo de un usuario (Nombre (s) + Apellido (s) o sólo Nombre (s)).

***Correo:*** este campo nos permite guardar el correo de un usuario.

***Tipo\_usuario:*** este campo define el tipo de usuario que se registra al sistema, es decir, “Tradicional”, “Oro”, “Platino”, “VIP”.

***Usuario:*** este campo define un nombre de usuario para el usuario que se registra al sistema, este nombre de usuario es único e irrepetible y sirve para búsqueda y recolección de información asociado al usuario que se registra en el sistema, a pesar de ser un campo único e irrepetible este campo no es la llave primaria.

***Pass:*** este campo almacena la información que hace referencia a una contraseña de seguridad para que el usuario pueda pasar el filtro de log in en el sistema.

### **Tabla Administradores.**

La tabla de administradores nos permite almacenar la información directa de un usuario administrador, en ella encontramos los siguientes campos.

***Id\_admin:*** este campo es de tipo entero, único e irrepetible y nos permite tener un registro único del administrador, al ser único e irrepetible es nuestra llave primaria.

***Nombre:*** este campo almacena el nombre completo de un administrador (Nombre (s) + Apellido (s) o sólo Nombre (s)).

***Direccion:*** este campo nos permite guardar la información de la dirección completa de un administrador (Calle + N. + CP + Colonia + Ciudad + Estado + País o de la forma más conveniente para el administrador)

***Telefono:*** este campo nos permite guardar la información del teléfono de un administrador.

***Correo:*** este campo nos permite guardar el correo de un administrador.

***Usuario:*** este campo define un nombre de usuario administrador para el usuario que se registra al sistema, este nombre de usuario es único e irrepetible y sirve para búsqueda y recolección de información asociado al usuario que se registra en el sistema, a pesar de ser un campo único e irrepetible este campo no es la llave primaria.

***Pass:*** este campo almacena la información que hace referencia a una contraseña de seguridad para que el administrador pueda pasar el filtro de log in en el sistema.

### **Tabla Películas.**

La tabla películas nos permite guardar la información de manera directa de una película, en esta tabla podemos encontrar los siguientes campos.

***Id\_pelicula:*** este campo es de tipo entero, único e irrepetible y nos permite tener un registro único asociado a una película, al ser único e irrepetible lo usamos como llave primaria.

***Nombre:*** este campo nos permite guardar la información del nombre de una película.

***Sipnosis:*** este campo nos permite guardar la información de una descripción breve de lo que se tratará una película.

***FLanzamiento:*** este campo nos permite guardar la información acerca de la fecha en que se originó o lanzó la película.

### **Tabla Géneros.**

La tabla géneros nos permite almacenar la información de un genero que se asocia a una película, nos sirve para la búsqueda específica de películas que se asocian a un género específico.

***Id\_genero:*** se opto por crear una clave para el genero para que todas las relaciones del sistema sean numéricas, al igual que los anteriores ID este campo es numérico, único e irrepetible y es una llave primaria.

***Género:*** este campo almacena el nombre de un género.

### **Tabla Salas.**

La tabla de salas nos permite almacenar la información de una sala que existente (físicamente) en el sistema, sus campos son.

***Id\_sala:*** este campo nos permite almacenar la clave de la sala, es de tipo entero, único e irrepetible, es una llave primaria.

***Numero\_sala:*** este campo nos permite almacenar información del nombre de una sala o en este caso un numero de sala, este campo es único e irrepetible.

***Tipo\_sala:*** este campo nos permite almacenar información del tipo de sala, es decir, “Tradicional”, “3D”, “VIP”.

***Numero\_asientos:*** este campo guarda la información del numero de asientos existentes en dicha sala, los asientos en este campo se representan por múltiplos de 8.

### **Tabla Asientos.**

La tabla de asientos contiene la información de todos los asientos que existen en el cine, el número total de asientos de esta representado por el número máximo de asientos que existen en una sala, es decir, si una sala tiene 24 asientos pero existe otra que tiene 32, el total de registros de asientos inscritos en la tabla asientos debe ser 32 para satisfacer la sala con 32 asientos.

***Id\_asiento:*** este campo nos permite identificar un asiento asignándole una clave, este campo es único e irrepitable y es la llave primaria de la tabla.

***Fila:*** este campo guarda la información de la fila en la que se encuentra el asiento, debe estar representada por letras.

***Asiento:*** este campo guarda la información del numero de asiento asociado a una fila, dicho campo debe ser numérico.

### **Tabla Horarios.**

La tabla horarios guarda la información de una fecha y una hora en la cual se proyectará una película en el cine.

***Id\_horario:*** este campo nos permite identificar un horario asignándole una clave, este campo es único e irrepitable y es la llave primaria de la tabla.

***Dia:*** este campo guarda la información del día en que se proyectará una película, el día está dado por el formato “AAAA-MM-DD”, es decir, que si se agrega un día debe estar dado por Año-Mes-Día puesto que es un campo de tipo date.

***Hora:*** este campo guarda la información de la hora en que se proyecta un horario, debe tener un formato de 24 hrs, sin embargo, depende de cómo lo maneje el administrador puesto que no hay restricciones.

### **Tabla Películas – Horarios.**

La tabla películas horarios relaciona las tablas películas y horarios de tal forma que a cada película le asigna un horario, esto se puede repetir, es decir que la película 1 se puede pasar en el horario 1 pero también en el 2 y el 3, al igual que el horario uno puede tener la película 2 y 3, las relaciones dependen del administrador.

***Id\_pelihorario:*** este campo es entero, único e irrepitable y nos sirve para identificar el registro de una relación película – horario en la tabla.

***Id\_pelicula:*** este campo es una llave foránea que asocia a una llave primaria de la tabla películas y nos regresa los datos de la clave a la que pertenece, es un entero.

***Id\_horario:*** este campo es una llave foránea que asocia a una llave primaria de la tabla horarios y nos regresa los datos de la clave a la que pertenece, es un entero.

### **Tabla Películas – Géneros.**

La tabla películas géneros relaciona las tablas películas y géneros de tal forma que a cada película le asigna un genero, esto no se puede repetir, la llave primaria de esta tabla es una llave compuesta por los id\_pelicula e id\_genero.

***Id\_pelicula:*** este campo es una llave foránea que asocia a una llave primaria de la tabla películas y nos regresa los datos de la clave a la que pertenece, sin embargo, también es una llave primaria para la tabla películas géneros.

***Id\_genero:*** este campo es una llave foránea que asocia a una llave primaria de la tabla géneros y nos regresa los datos de la clave a la que pertenece, sin embargo, también es una llave primaria para la tabla películas géneros.

#### **Tabla Películas – Salas.**

La tabla películas salas relaciona las tablas películas y salas de tal forma que a cada película le asocia una sala en la cual se transmitirá la película, al igual que en la tabla películas – horarios, esta tabla admite registros repetidos puesto que la película 1 se puede pasar en las salas 1, 2 y 3 al igual que la sala 1 también puede tener otra película en otro horario, todo depende de la forma en que lo maneje el administrador.

***Id\_pelicula\_sala:*** este campo es entero, único e irrepetible y nos sirve para identificar el registro de una relación película – sala.

***Id\_pelicula:*** este campo es una llave foránea que asocia a una llave primaria de la tabla películas y nos regresa los datos de la clave a la que pertenece, es un entero.

***Id\_sala:*** este campo es una llave foránea que asocia a una llave primaria de la tabla salas y nos regresa los datos de la clave a la que pertenece, es un entero.

#### **Tabla Salas – Asientos.**

La tabla salas – asientos relaciona las tablas salas y asientos de tal forma que a cada sala le asigna asientos respecto al número de asientos de la sala, es decir, que si la sala 1 tiene 24 asientos, la tabla salas asientos crea 24 registros donde a partir de la clave de la sala 1 le da los primeros 24 registros de los asientos registrados en la tabla asientos, de esta manera simulamos que la sala 1 tiene 24 asientos y cada asiento tiene una fila y un número de asiento.

***Id\_sala\_asiento:*** : este campo es entero, único e irrepetible y nos sirve para identificar el registro de una relación sala – asiento.

***Id\_sala:*** este campo es una llave foránea que asocia a una llave primaria de la tabla salas y nos regresa los datos de la clave a la que pertenece, es un entero.

***Id\_asiento:*** este campo es una llave foránea que asocia a una llave primaria de la tabla asientos y nos regresa los datos de la clave a la que pertenece, es un entero.

***Estado\_asiento:*** este campo nos permite conocer el estado en el que se encuentra un asiento en una sala, por ejemplo, el asiento ID:14 de la sala ID:2 se encuentra en buen estado, sin embargo el asiento ID:14 de la sala ID:3 no se encuentra en buen estado, es por eso que este campo esta guardado en tabla salas – asientos y no en la de asientos.

***Disponibilidad\_asiento:*** este campo nos permite conocer el estado en el que se encuentra un asiento en una sala, es decir, “disponible u ocupado” por ejemplo, el asiento ID:14 de la sala ID:2 se encuentra ocupado, sin embargo el asiento ID:14 de la sala ID:3 se encuentra disponible, es por eso que este campo esta guardado en tabla salas – asientos y no en la de asientos.

***Estado\_sala:*** este campo nos permite conocer las condiciones en las que se encuentra una fila en una sala, es decir la fila de asientos correspondiente a una sala, por ejemplo, la fila de asientos “A” de la sala 101 se encuentra en buen estado, sin embargo, la fila “B” en la misma sala no lo esta, lo llame estado sala puesto que quise hacer referencia a las palabras “estado en sala”.



## Tabla Tickets.

La tabla tickets nos sirve para relacionar las tablas películas – horarios y salas – asientos, ya que en este punto cada película tiene un horario y sala registrado y cada sala un asiento, es posible crear una conjunción de toda la información que corresponde a las tablas mencionadas y regresar un “ticket” que muestre la información relevante para el usuario final, dicha información contiene:

- ID ticket
- Nombre de la película.
- Día (Fecha).
- Hora.
- Numero de sala.
- Fila.
- Asiento.
- Precio.

Como podemos observar, la información anterior es el resultado de relacionar las tablas películas, horarios, salas, asientos y películas – horarios. Entonces podemos crear un resultado que contenga lo anterior mencionado.

***Id\_ticket:*** este campo es numérico, único e irrepetible, nos permite asignarle a un ticket una clave para identificar el registro entre películas – horarios y salas – asientos.

***Id\_pelihorario:*** este campo es una llave foránea que asocia a una llave primaria de la tabla películas - horarios y nos regresa los datos de la clave a la que pertenece, es un entero.

***Id\_sala\_asiento:*** este campo es una llave foránea que asocia a una llave primaria de la tabla salas - asientos y nos regresa los datos de la clave a la que pertenece, es un entero.

## Tabla Compras.

La tabla de compras nos permite finalmente simular una compra para el usuario final, esta tabla contiene el registro de un ticket creado así como un usuario existente, se pensó esta tabla con el fin de tener tickets y venderlos a 1 o mas usuarios, o que un usuario pueda comprar 1 o mas tickets, además de que es una gran herramienta para consultar que usuarios hicieron compras y de que tickets lo hicieron.

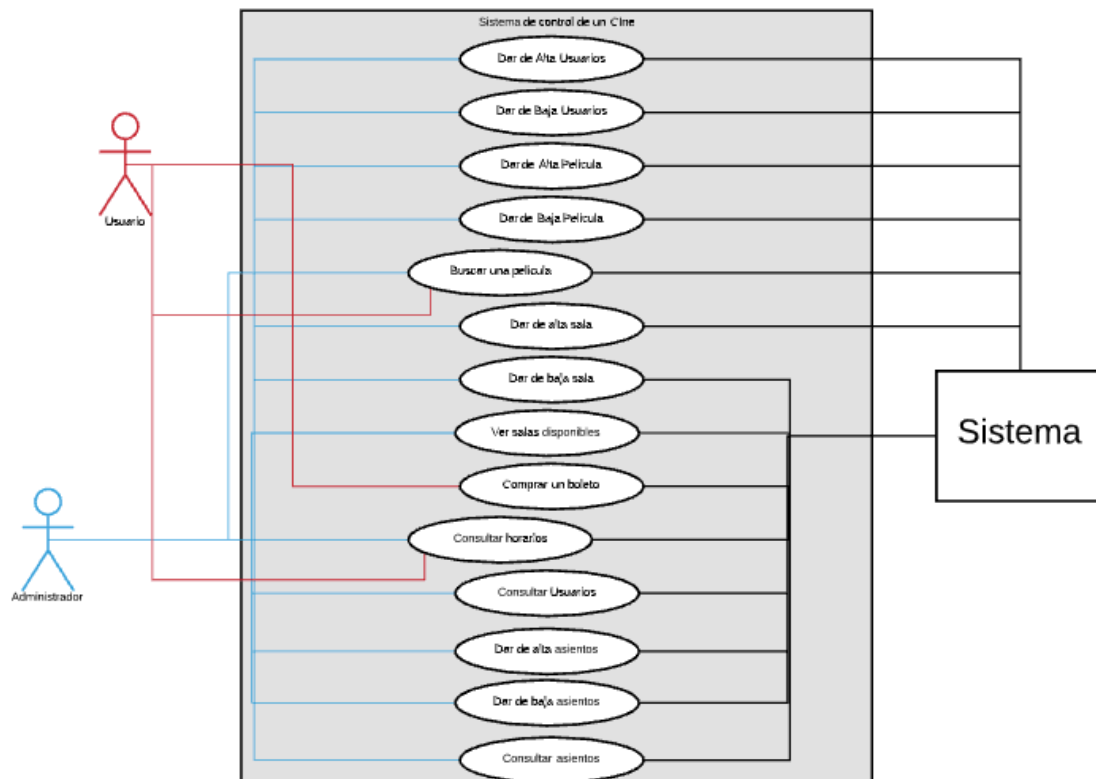
***Id\_compra:*** este campo es numérico, único e irrepetible, nos permite asignarle a una compra una clave para identificar el registro entre tickets y usuarios.

***Id\_ticket:*** este campo es una llave foránea que asocia a una llave primaria de la tabla tickets y nos regresa los datos de la clave a la que pertenece, es un entero.

***Id\_usuario:*** este campo es una llave foránea que asocia a una llave primaria de la tabla usuarios y nos regresa los datos de la clave a la que pertenece, es un entero.

Diagrama de casos de uso.

El diagrama de casos de uso muestra las actividades más relevantes que puede realizar el sistema de información.



**Figura 4.** Diagrama de casos de uso del sistema de información MVC de un cine.

Descripción de la actividad Comprar un boleto.

Caso de uso: Comprar un Boleto.	
Resumen:	Este caso de uso permite a un usuario registrado en el sistema comprar un boleto.
Actores:	Usuarios.
Fecha de creación:	27-Mayo-2020
Precondiciones:	El usuario esta logueado en el sistema, el usuario tiene un nombre de usuario y una contraseña
Post-condición.	Se puede consultar la informacion de su compra

<b>Escenario principal:</b>	<ol style="list-style-type: none"> <li>1. El usuario se encuentra en el menú principal y observa la opción correspondiente a comprar un boleto, ingresa el número y da enter.</li> <li>2. El sistema pregunta su nombre de usuario.</li> <li>3. El usuario ingresa su nombre de usuario y da enter.</li> <li>4. El sistema verifica su nombre de usuario, posteriormente pregunta el nombre de la película.</li> <li>5. El usuario ingresa el nombre de la película y da enter.</li> <li>6. El sistema muestra la cartelera con las claves asociadas al nombre que ingreso el usuario.</li> <li>7. El usuario observa la clave de la película que quiere ver, ingresa el número correspondiente y da enter.</li> <li>8. El sistema muestra las salas con las claves asociadas a la clave que selecciono el usuario.</li> <li>9. El usuario ingresa la clave de la sala en la que quiere ver la película.</li> <li>10. El sistema le pide al usuario seleccionar un asiento en la sala que eligió.</li> <li>11. El usuario ingresa los valores correspondientes a la fila y asiento y da enter.</li> <li>12. El sistema verifica que todas las claves están correctas, y pide confirmar la compra con su contraseña.</li> <li>13. El usuario ingresa su contraseña para confirmar la compra.</li> <li>14. El sistema verifica que su contraseña sea correcta y confirma la compra, le muestra al usuario las opciones de ver detalles de la compra con Y/N</li> <li>15. El usuario selección Y o N y da enter.</li> <li>16. Conforme a la opción ingresada el sistema muestra o no los detalles de la compra, y vuelve al menú principal.</li> </ol>
<b>Flujos alternativos:</b>	<p>A1: EL USUARIO NO EXISTE: Aquí el sistema debe reportar al usuario que el nombre de usuario que ingreso no existe.</p> <p>A2: LA PELICULA NO EXISTE: Aquí el sistema debe reportar al usaurio que la película no existe.</p> <p>A3: LA SALA NO EXISTE: Aquí el sistema debe notificar al usaurio que la sala no existe.</p> <p>A4: EL ASIENTO ESTA OCUPADO O NO EXISTE: Aquí el sistema debe notificar al usuario que el asiento NO EXISTE</p> <p>A5: LA CONTRASEÑA NO ES IGUAL: Aquí el sistema debe notificar al usuario que la contraseña que ingreso para confirmar la compra no es igual a la registrada en la base de datos.</p>

En conclusión, aprendimos que para realizar un sistema de información y en general cualquier aplicación del tipo modelo – vista – controlador, es necesario entender los fundamentos básicos a lo que esto se refiere y empezar estructurando un diseño simple como lo es el modular para poder crear un modelo relacional que ya incluya relaciones y llaves primarias o foraneas asociadas a un campo, también que no siempre es necesario normalizar un modelo relacional, es una buena práctica académica, sin embargo no siempre es necesario puesto que depende mucho del contexto para el cual se normaliza.

En la creación de la base de datos nos dimos cuenta que una vez que se entiende el comportamiento de la información es muy fácil crear la estructura de la base de datos, además que teniendo el modelo relacional creamos una base para la codificación en el manejador de bases de datos, y una vez creada la base de datos poder empezar la codificación en el lenguaje de programación.