



# PRACTICA INDIVIDUAL

## Implementación de un sistema para comprar un boleto de cine

### Descripción breve

El sistema debe ser para la compra de un boleto de cine. Donde exista un usuario y un administrador, donde el usuario tiene acceso a ver los horarios de las películas y la compra de boletos. Mientras que el administrador se encarga de administrar el sistema.

Jorge de Jesús Manriquez Martinez 145664

Universidad de Guanajuato

# Sistema para la compra de un boleto de cine

## **Objetivos**

### **Objetivo General**

Se realizará un sistema para la compra de un boleto de cine con el fin de sistematizar el proceso de las funciones CRUD (Create, Read, Update and Delete), para cada uno de los atributos que se implementen. También tener algunas otras acciones como comprar un boleto. Se realizará mediante las herramientas vistas en clase como lo son diagrama modular, diagrama casos-usos, diagrama relacional con las tres normalizaciones. El desarrollo será mediante MySQL y Python.

### **Objetivos Específicos**

- Administrar administradores, usuarios, películas, salas, horarios (funciones), asientos, boletos, tickets.
- Comprar boleto

### **Descripción del problema**

Realizar un sistema para la compra de un boleto de cine, donde se automatice el manejo de la información de administradores, usuarios, películas, salas, horarios (funciones), asientos, boletos, tickets, así como también la compra de boletos. Este manejo de información será mediante crear, leer, actualizar y borrar, cada una de estas funciones se aplicará para los atributos mencionados anteriormente.

### **Requisitos de información**

1. Datos del administrador incluyendo un usuario y contraseña.
2. Datos de usuarios(opcional)
3. Datos películas
4. Datos salas
5. Datos horarios
6. Datos asientos

7. Datos boletos
8. Datos tickets

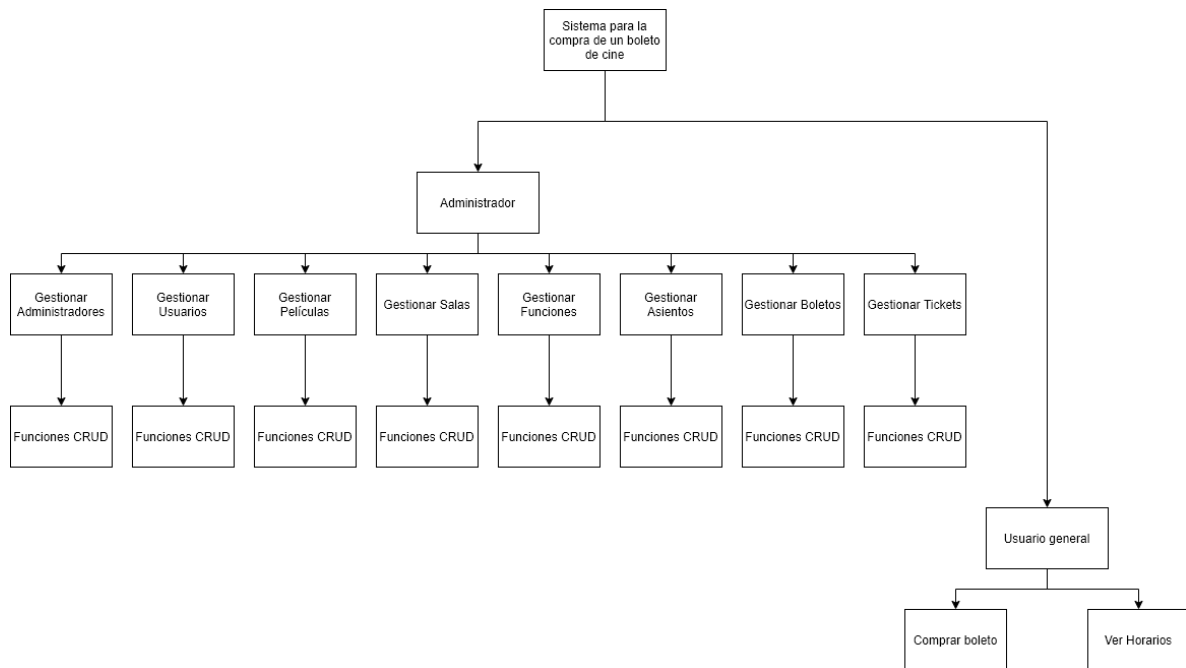
### **Requisitos de procesos**

1. *Validar si eres usuario general o administrador.* Aquí se selecciona que tipo de usuario eres. Si eres usuario general se despliega un menú donde te da las opciones de ver todos los horarios, horarios por nombre de la película o por fecha y comprar un boleto. Mientras que, si eres administrador valida con un usuario y contraseña, una vez que se ingrese se despliega un menú donde puedes gestionar cada uno de los atributos implementados.
2. *CRUD administradores.* Te permite crear, leer de distintas maneras, actualizar y borrar.
3. *CRUD usuarios.* Te permite crear, leer de distintas maneras, actualizar y borrar.
4. *CRUD películas.* Te permite crear, leer de distintas maneras, actualizar y borrar.
5. *CRUD salas.* Te permite crear, leer de distintas maneras, actualizar y borrar.
6. *CRUD horarios.* Te permite crear, leer de distintas maneras, actualizar y borrar.
7. *CRUD asientos.* Te permite crear, leer de distintas maneras, actualizar y borrar.
8. *CRUD boletos.* Te permite crear, leer de distintas maneras, actualizar y borrar.
9. *CRUD tickets.* Te permite crear, leer de distintas maneras, actualizar y borrar.
10. *Comprar un boleto.* Te permite crear, leer de distintas maneras, actualizar y borrar.

### **Diseño conceptual**

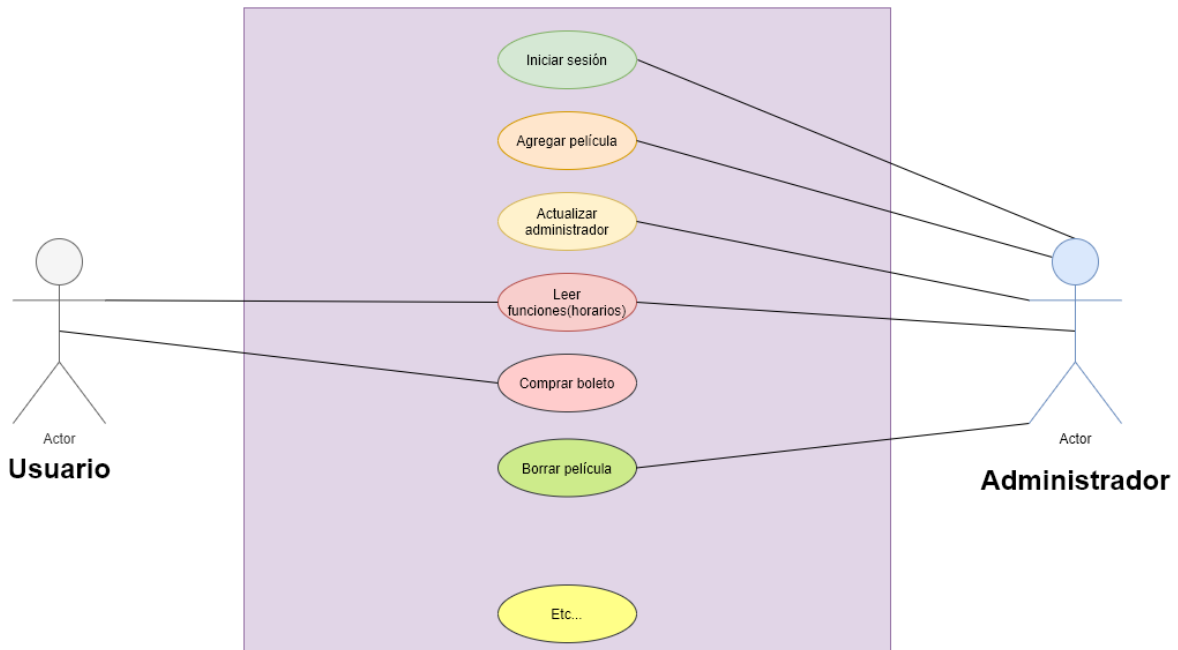
#### **Diagrama Modular**

A continuación, se muestra el diagrama modular donde se presentan cada uno de los módulos con los que cuenta el sistema y con las operaciones que puede hacer cada módulo.



## Diagrama de casos-usos

### Sistema para la compra de un boleto de cine



## Actividad para describir

**Nombre:** Agregar película

**Suposiciones:**

- Dado de alta en el sistema

**Precondiciones:**

- Inicio de sesión
- Este en el menú adecuado

**Inicio:**

- Seleccionar la opción adecuada para agregar película.

**Flujo de eventos:**

*Sistema:* Muestra menú dos opciones (usuario general y administrador).

*Usuario:* Selecciona la opción “administrador”.

*Sistema:* Muestra menú con todos los submenús.

*Usuario:* Selecciona la opción “Películas”.

*Sistema:* Muestra el submenú de Películas con las operaciones que se pueden realizar.

*Usuario:* Selecciona la opción “Agregar película”.

*Sistema:* Pide los datos para satisfacer esta acción.

*Usuario:* Ingresa los datos.

*Sistema:* Arroja un mensaje (agrego, no agrego).

*Usuario:* Sale del sistema.

**Diagrama relacional**

A continuación, se muestra el diagrama relacional para el sistema sin ninguna forma normal. Este diagrama muestra todas las tablas que se implementaran en nuestra base de datos.

Contiene una tabla “películas” con sus campos que se requirieron adecuados. Esta relacionada con otra tabla “función”, esta representa los horarios de las películas, por lo cual están relacionadas entre ellas, la relación es de 1 a muchos, ya que una película puede estar en cero o muchas funciones, pero una función solo puede transmitir una película por función.

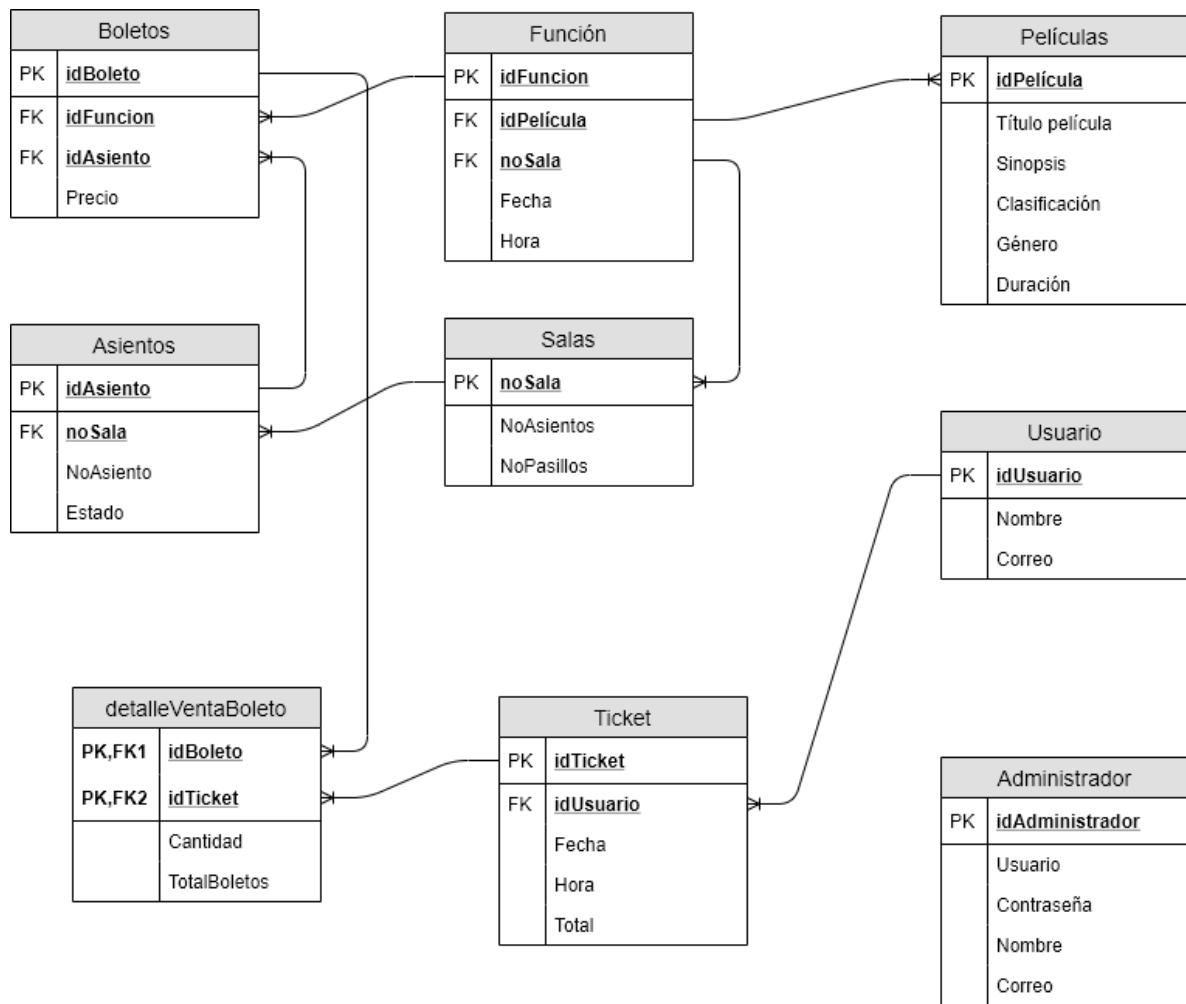
La tabla “función” tiene otras dos relaciones, una con la tabla “salas”, la cual representa las salas que existen en el cine. Estas tablas contienen una relación de 1 a muchos, ya que una sala puede proyectar muchas funciones, pero una función solo puede ser proyectada en una sala. Mientras que la tabla “salas” tiene otra relación con la tabla “Asientos”, dicha relación es uno 1 a muchos, ya que una sala puede tener muchos asientos, pero un asiento solo puede estar en una sala.

La tabla “Boletos”, la tabla boletos contiene varia información de otras tablas, por lo cual tiene varias relaciones. Una de ellas es con la tabla “función”, donde su relación es 1 a muchos, porque un boleto solo puede tener cero o una función, pero una función puede tener cero o muchos boletos.

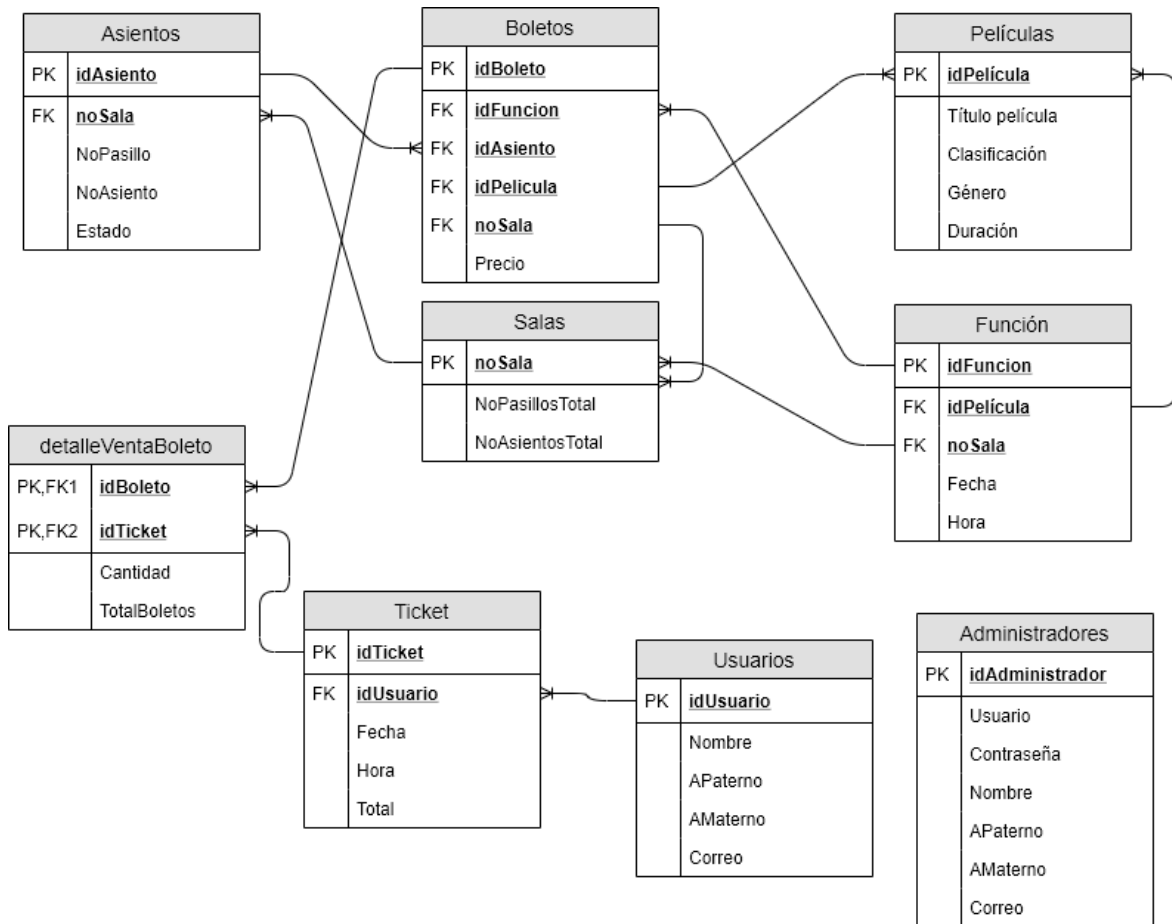
La tabla “Asientos” contiene la información del asiento, número de asiento y su estado (ocupado, libre), tiene una relación con la tabla “Boletos”, dicha relación es 1 a muchos, porque un boleto solo puede tener un número de asiento, mientras que un asiento puede tener muchos boletos.

La tabla “Boletos” y la tabla “ticket” tenían una relación muchos a muchos ya que un boleto podía aparecer en muchos tickets y a la vez muchos tickets podían contener un boleto, por lo cual se decidió hacer una tabla para que uniera estas dos con relaciones 1 a muchos.

La tabla “administradores”, la cual no tiene ninguna relación con ninguna otra tabla, pero como para este caso en particular, no es necesario que esta este relacionado con alguna otra.

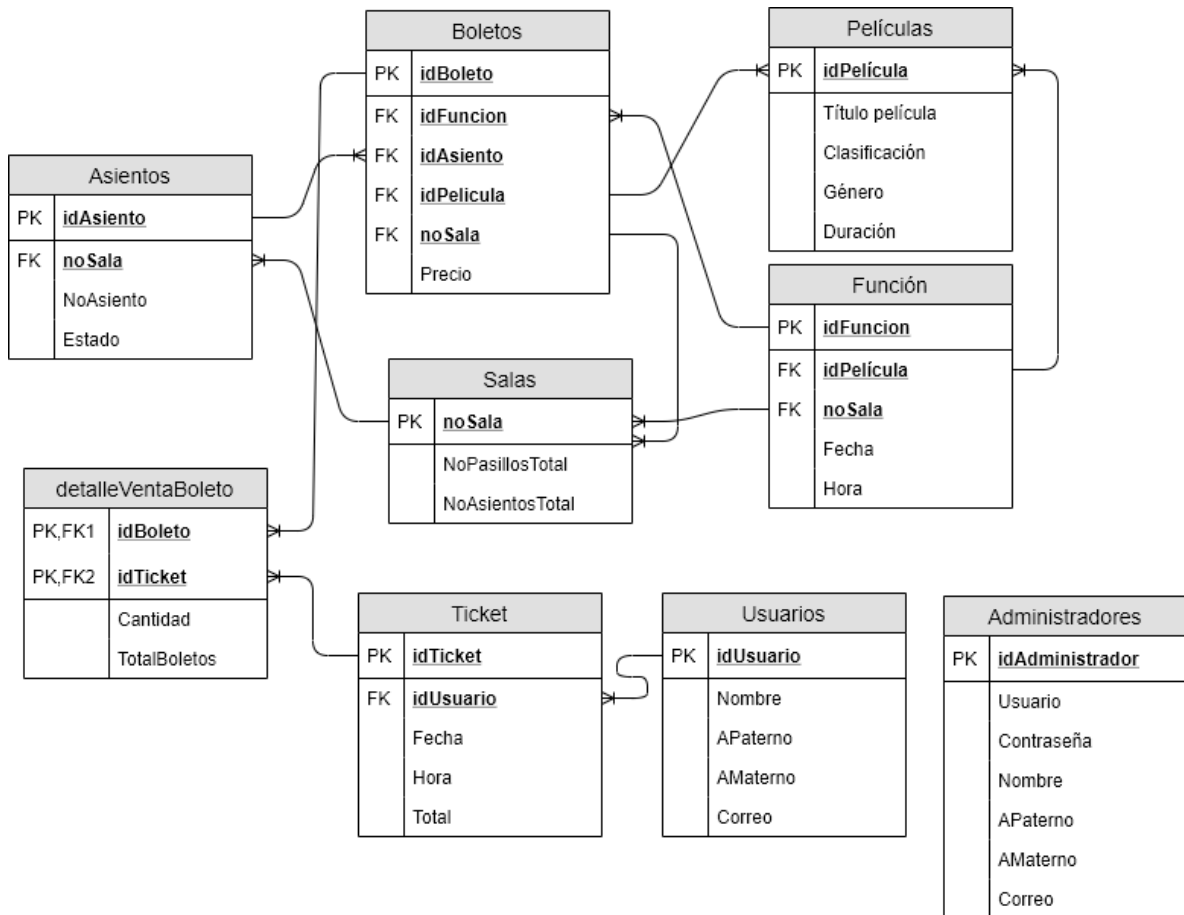


Una vez que se implemento el diagrama relacional anterior, se le aplico la primera forma normal (1NF), aplicando esta quedo de la siguiente manera:



Para la segunda forma normal (2NF) no se realizó ningún cambio ya que cumplió la condición dos, que nos dice que cada columna que no es una clave depende completamente de la clave primaria, por lo cual quedo de la misma manera ya que ninguna columnas dependían de la clave a excepción de las claves foráneas.





Para la tercera forma normal (3NF) si hubo cambios, ya que esta no cumplió con la segunda condición, esta dice que no que no hay dependencia funcional transitiva, es decir, todas las columnas que no son parte de la clave son independientes entre ellas, aplicando la 3NF quedo como se muestra a continuación:

