



Campus Irapuato-Salamanca | División de Ingenierías



UNIVERSIDAD DE
GUANAJUATO

UNIVERSIDAD DE GUANAJUATO
DIVISIÓN DE INGENIERÍAS, CAMPUS IRAPUATO-SALAMANCA

Sistemas de Información.
Dr. Juan Carlos Gómez Carranza.

Práctica Individual.

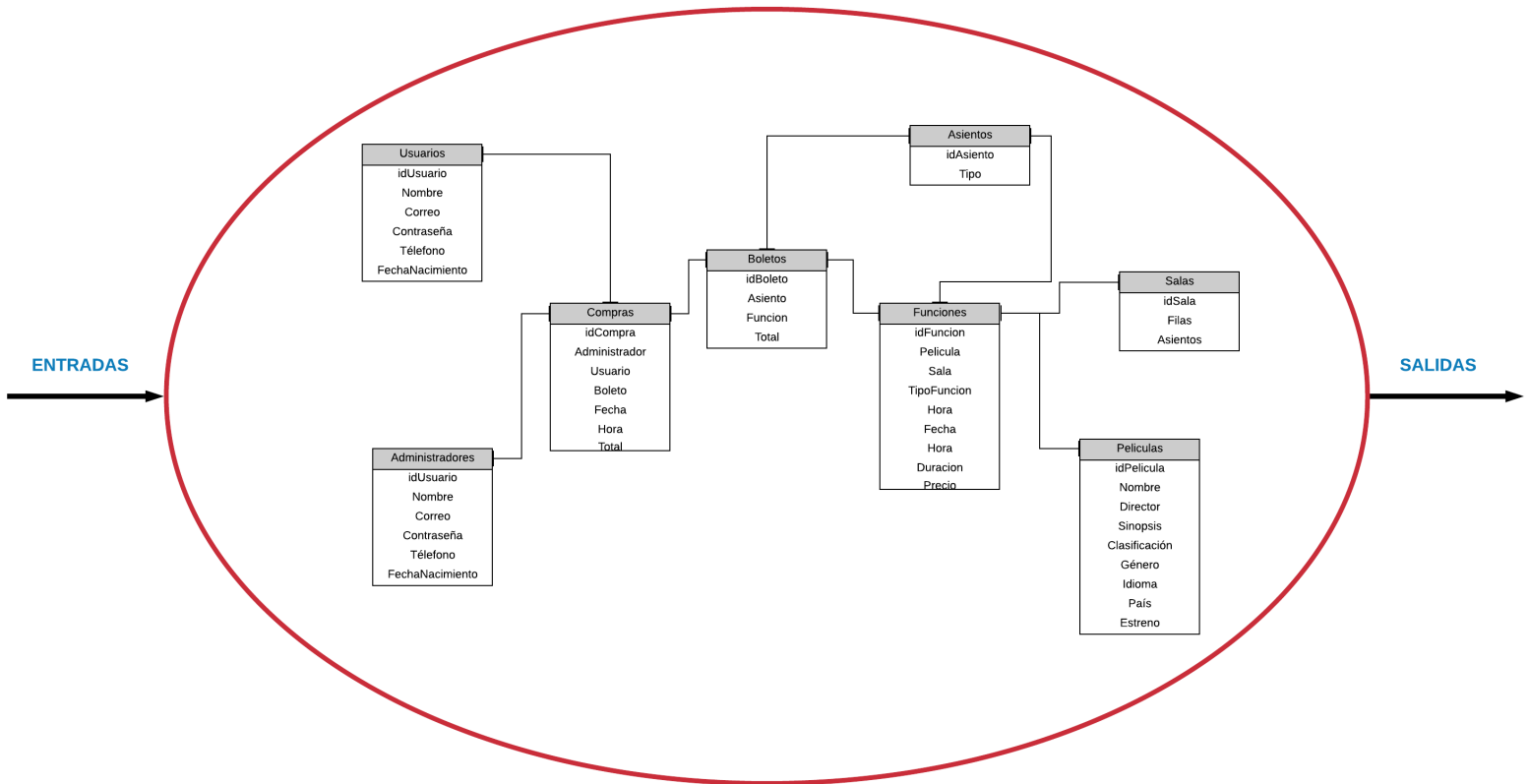
Jonathan Joel Corona Ortega.
NUA: 780840

Ingeniería en Sistemas Computacionales.

Fecha de Entrega:
29 de mayo del 2020.

DISEÑO MODULAR.

Sistema de un cine.



Ambiente: Cinema.

Actores:

Usuarios

- Clientes.
- Administradores.

Entradas:

- Administrador:
 - Su información (Registro o Login): Nombre, correo, contraseña, teléfono, cumpleaños.
 - Información de películas: Título, director, sinopsis, clasificación, género, entre otros.

- Información de salas: Número de asientos y de filas.
- Información de los asientos: Enumeración, tipo de asiento
- Información de las funciones: Película, sala, tipo de función, hora, fecha, duración, precio.
- Información del boleto: Asiento a asignar a una función.
- Compras.
- Cliente:
 - Su información (Registro o Login): Nombre, correo, contraseña, teléfono, cumpleaños.
 - Compras.

Salidas:

- Administrador:
 - Información nueva añadida a las tablas del sistema.
 - Información editada en las tablas del sistema, ya sea actualizada o eliminada.
 - Ventas de boletos.
 - Información de la compra que realizaron.
- Cliente:
 - Información de sus boletos.
 - Información de la compra que realizaron.

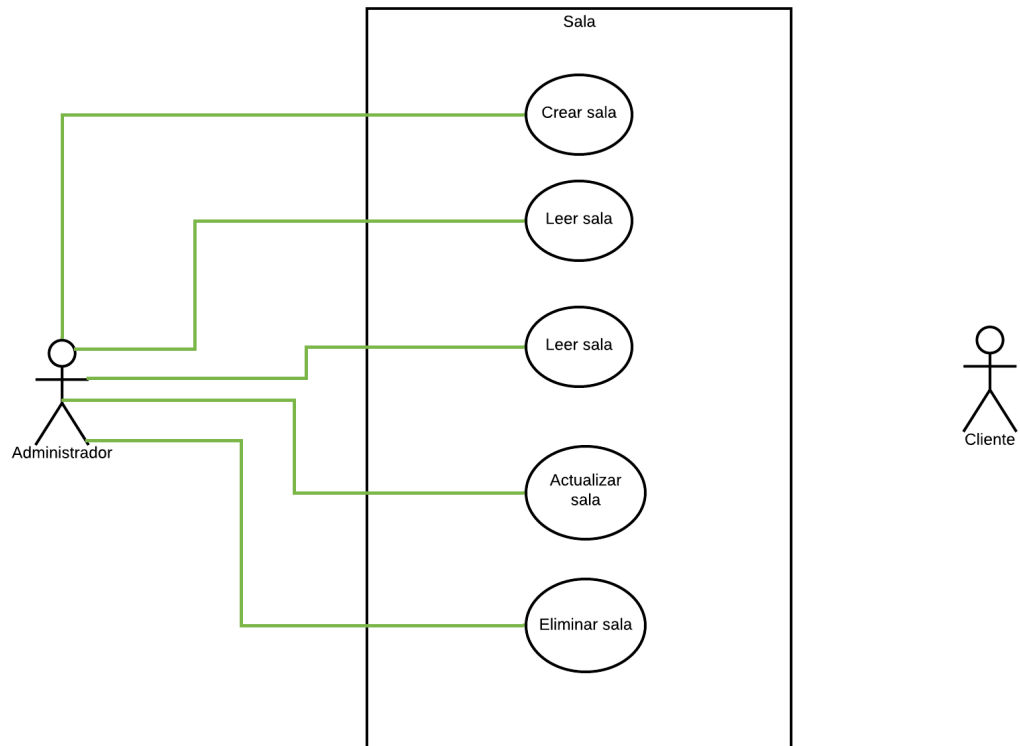
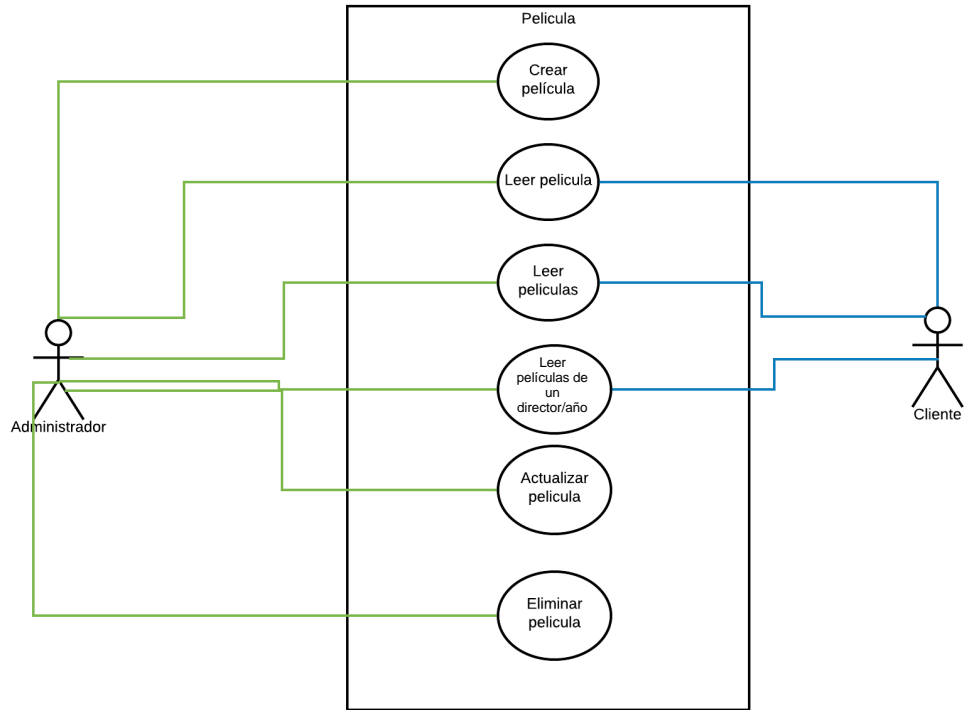
Información de acción.

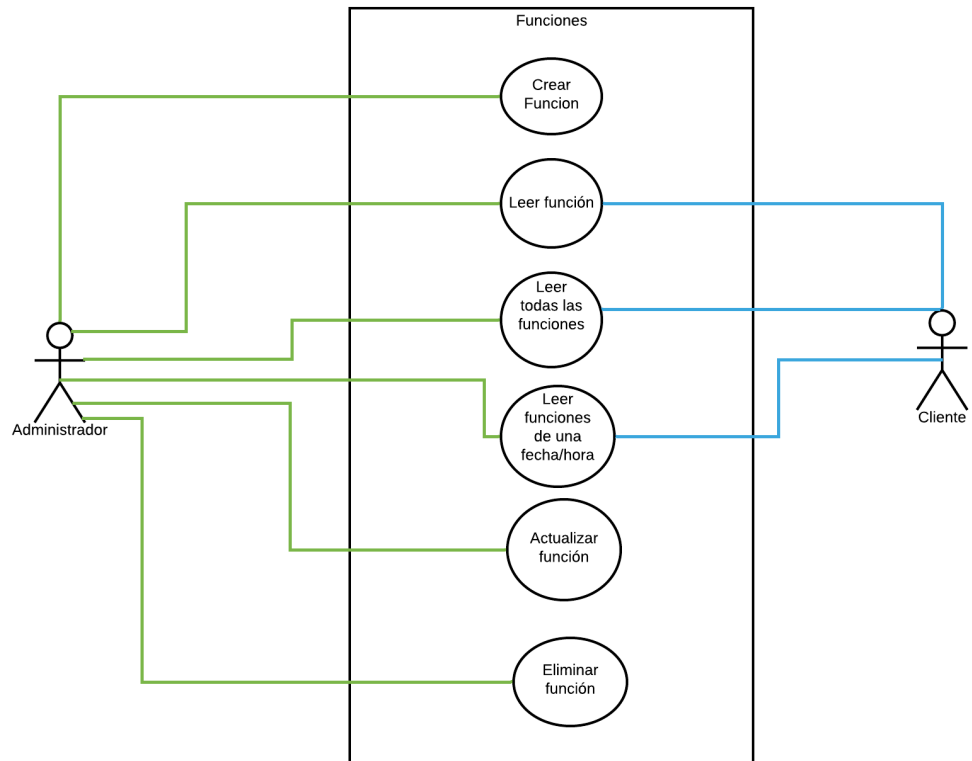
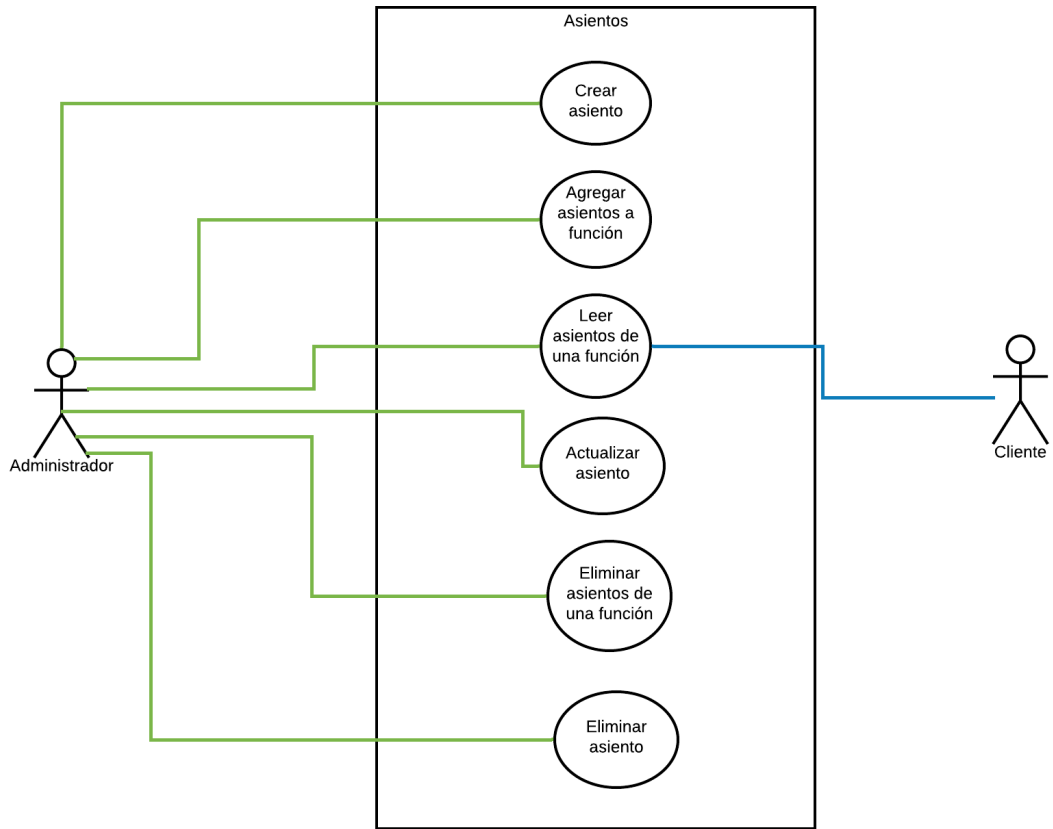
- Dispositivos de entrada:
 - Clic(mouse).
 - Teclado
- Dispositivos de salida.
 - Pantalla.
 - (Opcional) Impresora: Impresión de boletos

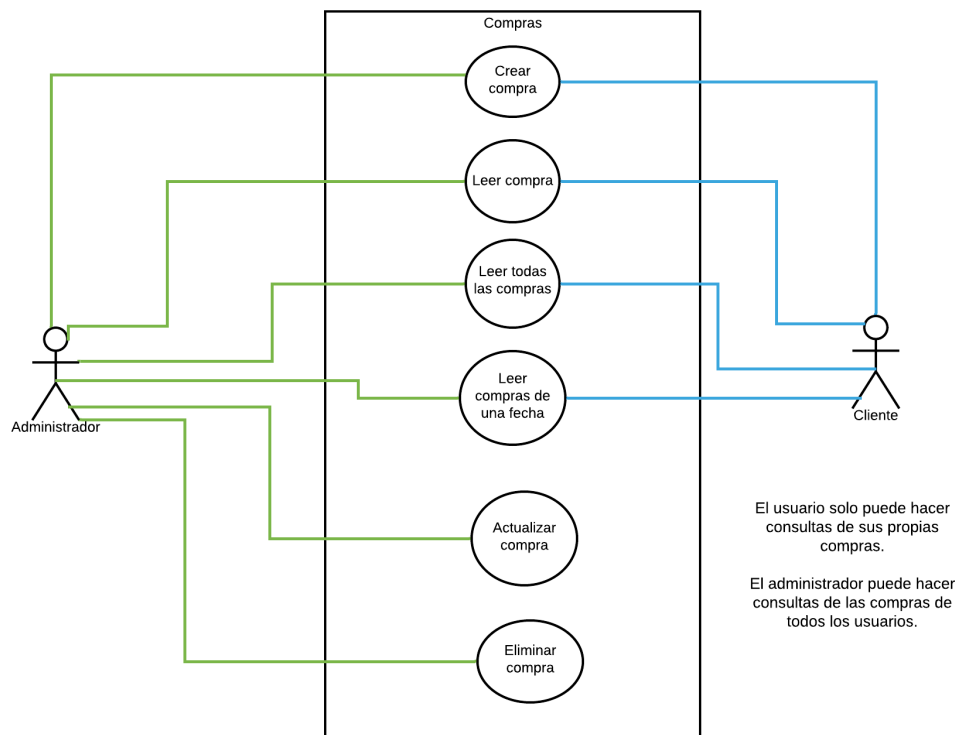
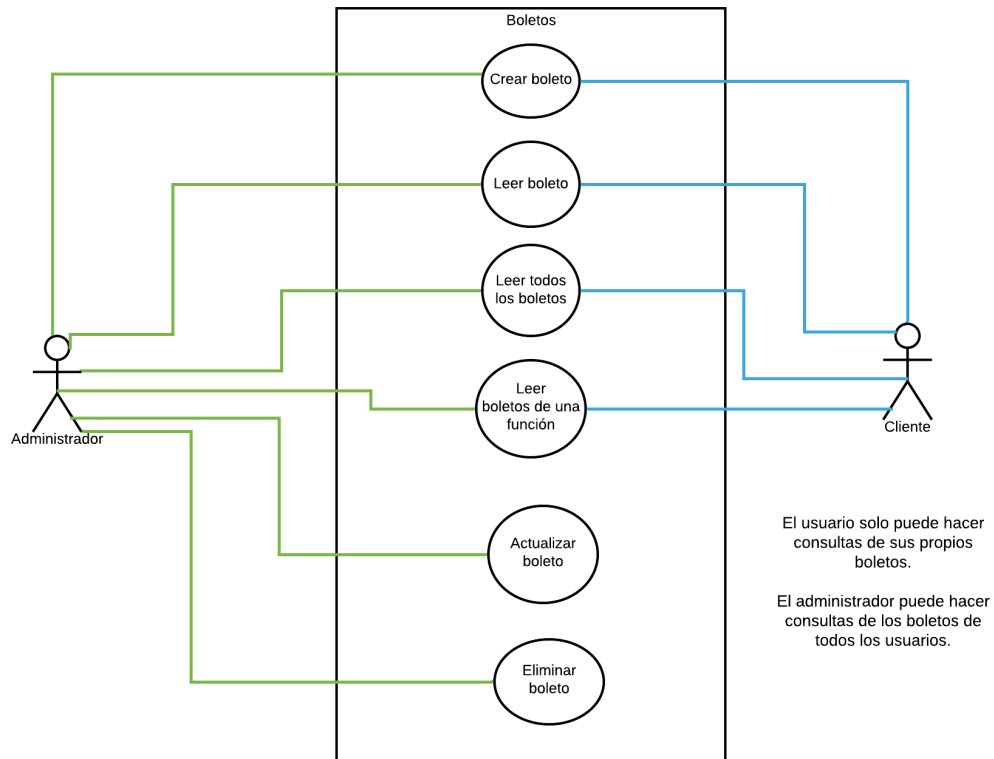
Plataforma.

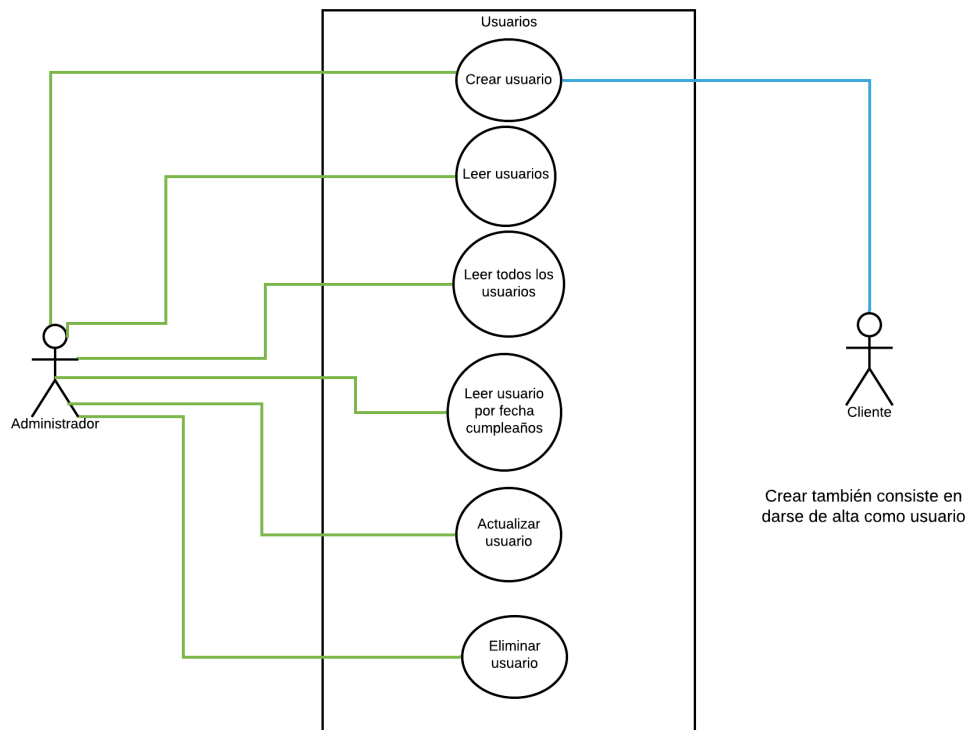
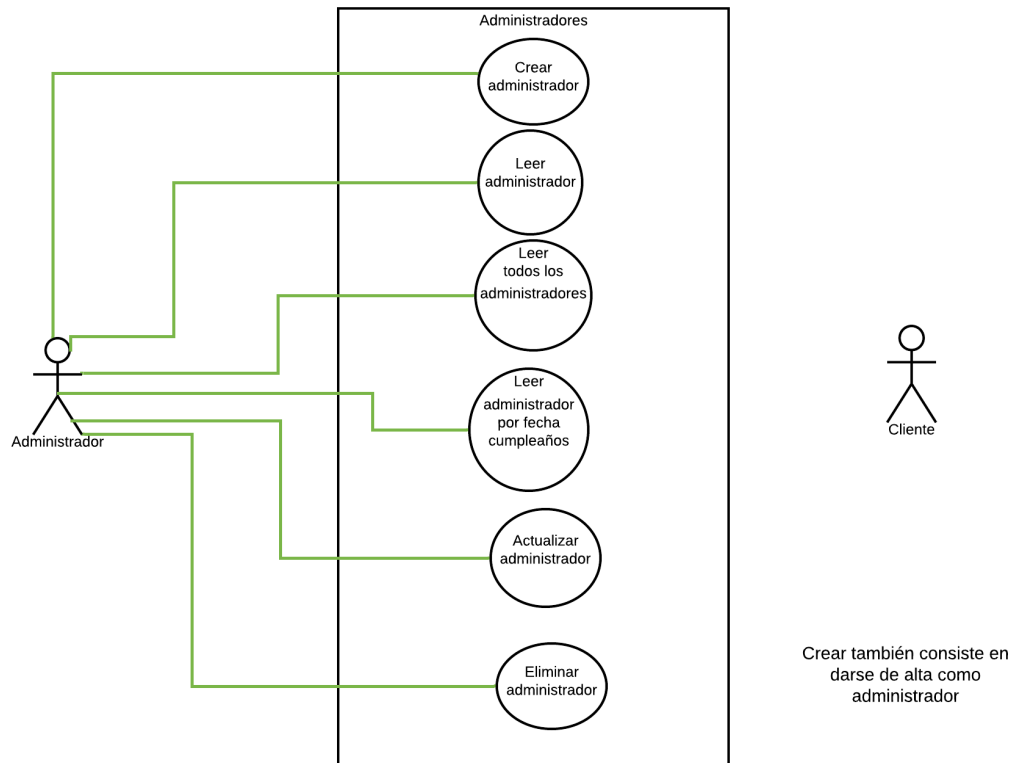
- Actual: Escritorio.
- Posibles: Web y/o móvil.

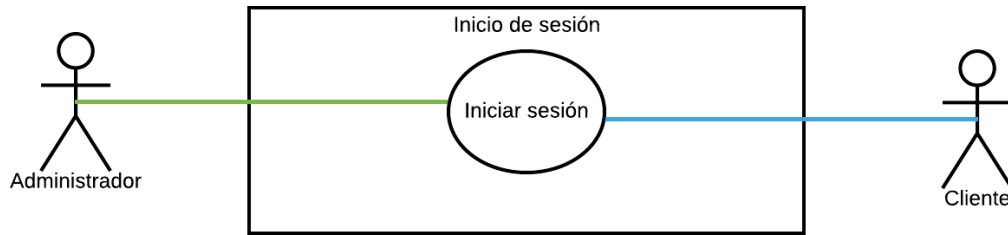
CASOS DE USO.











CASOS DE USO: BOLETOS.

Nombre del caso de uso: Crear boleto (Administrador).

Suposiciones:

- Usuario dado de alta.
- Haber iniciado sesión.
- Funciones creadas o disponibles.
- Haber ingresado asientos.

Pre-condiciones:

- Haber ingresado en el submenú de compras.
- Haber ingresado la opción de Boletos del submenú de compras.
- Contar asientos disponibles en la función.

Inicio:

- Salida → Muestra todas las opciones de los Boletos.
- Entrada → Clic en Crear boleto.

Flujo de eventos:

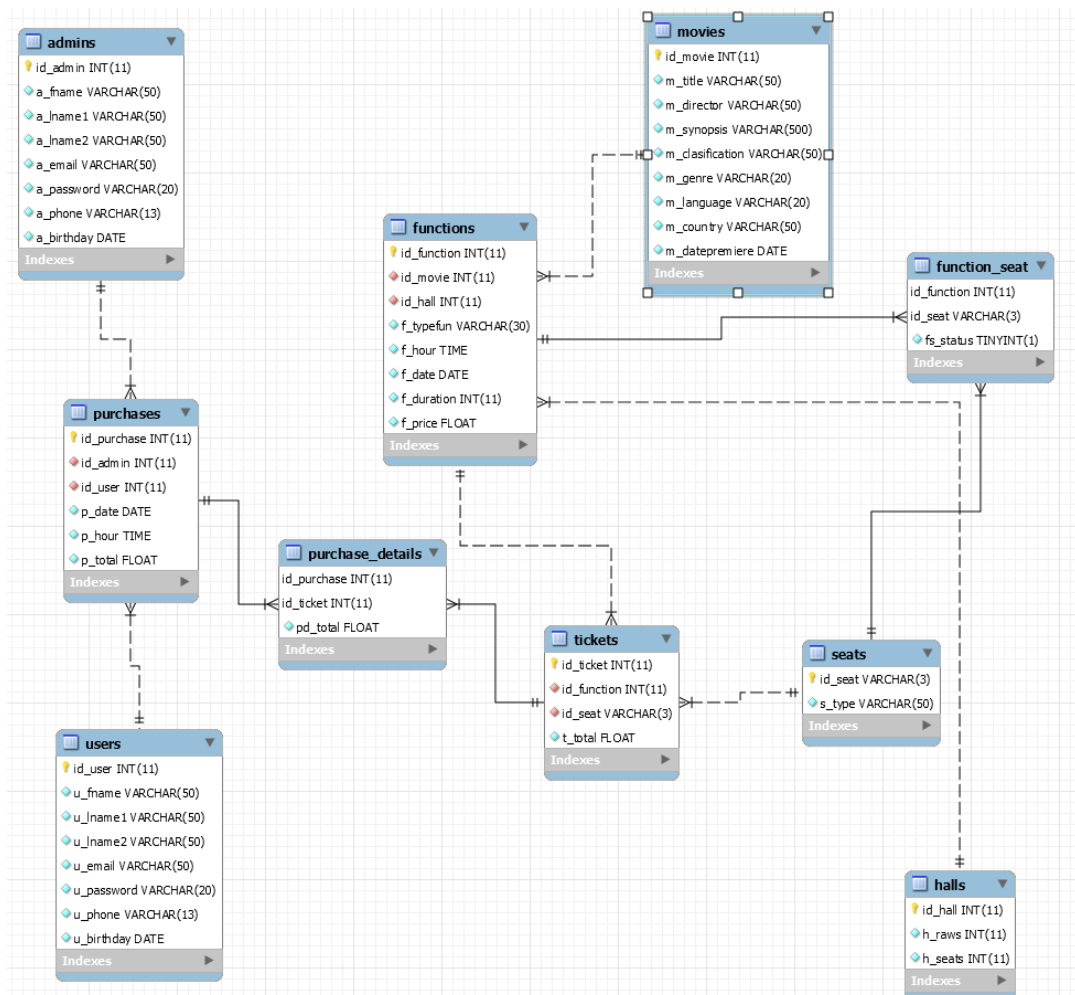
- Salida → Despliega pregunta de cuál función se desea ver con asientos disponibles.
- Entrada → Ingresas el número de función.
- Salida → Despliega el número de función, los asientos y su disponibilidad
- Salida → Pregunta de nuevo la función a la que quieres entrar.
- Entrada → Ingresas de nuevo la función que se desea.
- Salida → Pregunta el número de asiento que desees.
- Entrada → Ingresas el asiento que se desea.

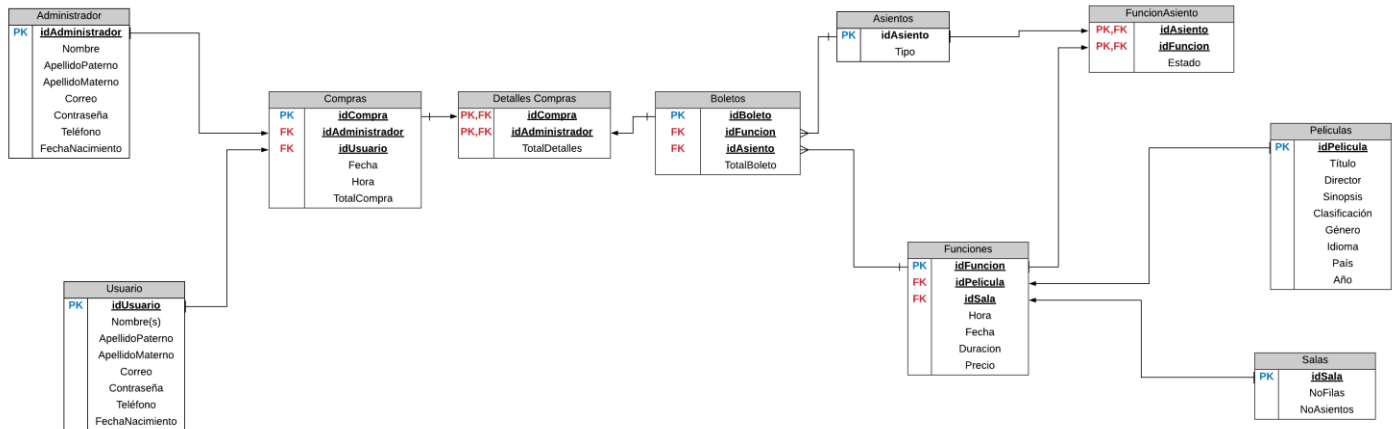
- Salida → Despliega una notificación o mensaje que indica que la reservación se realizó correctamente.
- Salida → Muestra la función y el asiento reservado.
- Salida → Regresa al menú de Boletos dentro del submenú de compras.
- Si se desea comprar otro boleto, se tiene que dar clic de nuevo en Crear Boleto (Se repite el mismo flujo de eventos).

Post-condiciones:

- Se actualiza o cambia la base de datos:
 - El estado de los asientos de una función cambia de disponible a ocupado/reservado.
 - La tabla Boletos presenta un nuevo registro.

DISEÑO DEL DIAGRAMA DE BASE DE DATOS.





Relaciones.

- Un administrador puede tener muchas compras, con diferentes clientes, pero una compra no puede tener varios clientes ni administradores. (Uno a muchos).
- Una compra puede tener muchos boletos, pero un boleto(id) no puede estar en varias compras. (Uno a muchos).
- Un boleto no puede tener muchos asientos, ni muchas funciones. Pero una función si puede tener muchos boletos. (Uno a muchos).
- Un asiento(id) puede tener muchas funciones y una función puede tener muchos asientos. (Muchos a muchos).
- Un función solo puede tener una sala, pero una sala puede tener varias funciones. (Uno a muchos).
- Una función solo puede tener una película, pero una película puede estar en varias funciones diferentes.

Normalización.

- 1era Formal Normal (1NF):
 - En cada tabla, cada columna contiene valores atómicos, a excepción de la fecha, debido a que MySQL maneja este tipo de datos (DATE) lo que permite utilizar las fechas completas, y si deseamos obtener algún valor específico, como el año, ya hay funciones que permiten hacer eso.
 - Cumple porque todos los valores en una columna son del mismo dominio.
 - Cumple porque todas las columnas de una tabla tienen nombres únicos.
 - No hay orden en columnas y filas.
 - Cumple porque todas las columnas son regulares.

- 2da Forma Normal (2NF):
 - La tabla está en 1NF.
 - Cumple porque en cada columna que no es una clave depende completamente de la clave primaria (incluso si es una clave compuesta).
- 3ra Forma Normal (3NF):
 - La tabla está en 2NF.
 - Cumple porque no hay dependencia funcional transitiva, es decir, todas las columnas que no son parte de la clave dependen exclusivamente de la clave primaria y son independientes entre ellas.

Conclusión.

Esta práctica se me complicó más en la parte del diseño de la base de datos, debido a que, en un principio, no sabía como añadir los asientos a una función y como actualizarlos, pero al final lo pude resolver con consultas sencillas de MySQL. Tuve que repasar algunas cosas teóricas y prácticas sobre bases de datos, gracias a esto me permitió desarrollar el sistema de manera más rápida. Tal vez algunas cosas se pudieron realizar de manera diferente, pero en mi caso, creo que fue la manera más sencilla.

Esta práctica me ayudó mucho para reconocer mis virtudes y deficiencias en el ámbito de programación y de diseños de sistemas. En algunas partes fue algo frustrante, pero todo se puede resolver de acuerdo con lo establecido.