



Universidad de Guanajuato

División de Ingenierías Campus Irapuato-Salamanca

Ingeniería en sistemas

Sistemas de información

Dr. Juan Carlos Gómez Carranza

Práctica: Sistema de un cine

Oscar Montes Camberos

5 de mayo del 2019

INTRODUCCIÓN:

A lo largo de esta práctica se diseñó e implementó un sistema de un cine, en el cual los usuarios pueden comprar boletos para las diferentes funciones, todas éstas creadas previamente por un administrador.

DIAGRAMA MODULAR:

Plataformas: Escritorio

Ambiente: Cine

Actores: Administrador y usuario normal

Entradas:

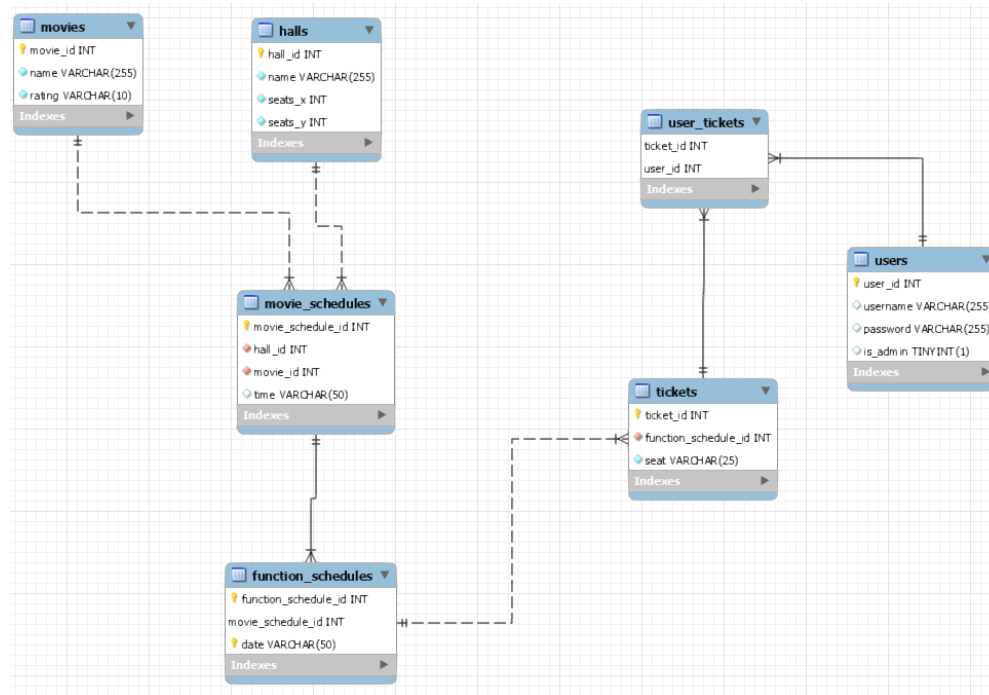
Información del usuario:

- ID de usuario
- Contraseña
- Información de salas, horarios, películas (en caso de ser administrador)

Información de acción:

- Teclado

Diagrama:

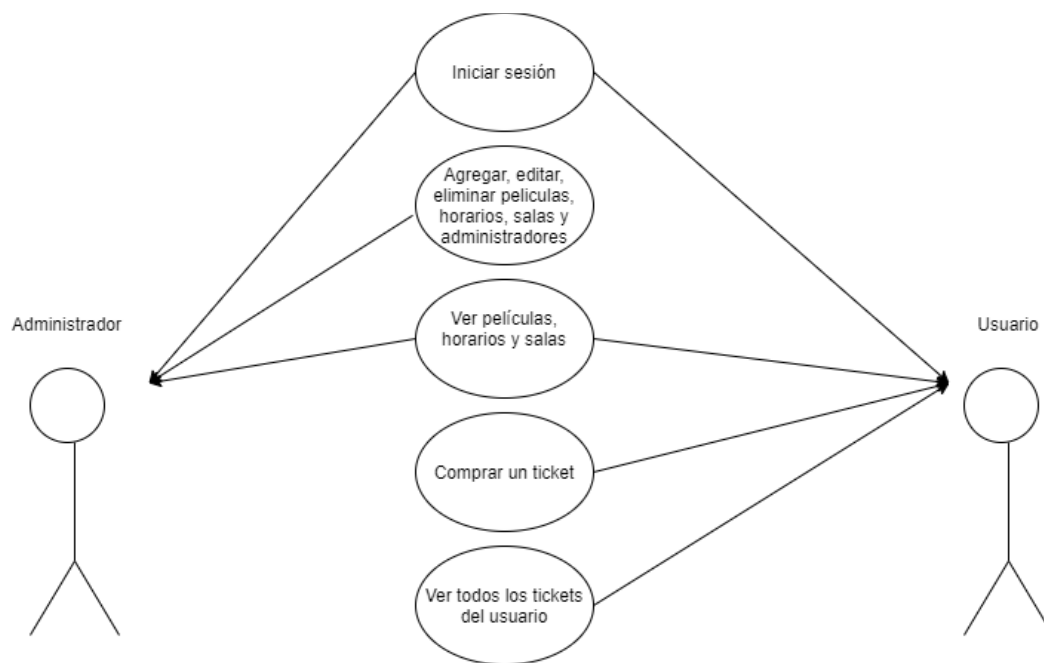


Salidas:

Usuario:

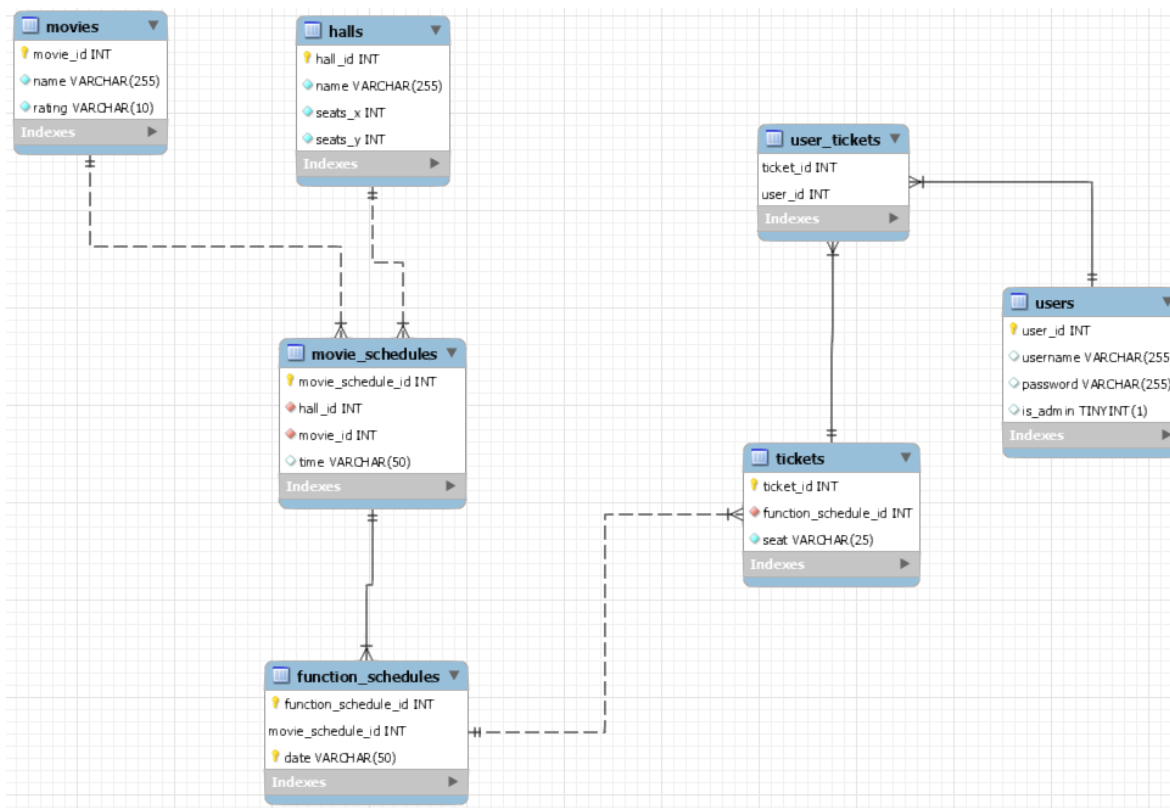
- Ticket de compra.

DIAGRAMA CASOS DE USO:



Caso de uso	Ver todos los tickets del usuario
Actor	Usuario
Descripción	Permite visualizar los tickets comprados por un usuario en específico
Precondiciones	El usuario ya se ha iniciado sesión con un ID de usuario y contraseña
Postcondiciones	Se mostró con éxito los tickets comprados por el usuario
Suposiciones	El usuario tiene un ID de usuario y contraseña
Pasos a realizar:	
<ol style="list-style-type: none"> 1. El usuario solicita al sistema comenzar con el proceso de mostrar los tickets comprados por el usuario. 2. El sistema busca los tickets comprados en la base de datos según el ID de usuario y los muestra al cliente en la pantalla. 	

DIAGRAMA RELACIONAL SQL:



SNIPPETS DE CÓDIGO IMPORTANTE:

Se incluyen a continuación fragmentos del código importantes:

Comprar un ticket:

```
def buy_ticket_menu(self):
    movies = self.model.get_schedules()
    self.view.show_movies(movies)
    self.view.msg('Introduce el numero de la pelicula para la que deseas comprar el boleto:')
    number_selected = input()
    movie_selected = movies[int(number_selected)-1][0]

    self.view.msg('Introduce el numero del horario en la que quieres reservar:')
    movie_schedules = movies[int(number_selected)-1][2]
    self.view.show_movie_schedules(movie_schedules)
    number_selected = input()
    schedule_selected = movie_schedules.split(',')[int(number_selected)-1]

    self.view.msg('Introduce el numero de la fecha en la que quieres reservar:')
    next_days = self.model.get_next_days()
    self.view.show_next_days(next_days)
    number_selected = input()
    date_selected = next_days[int(number_selected)-1]

    self.view.msg('Introduce el asiento que quieres reservar, ejemplo: B6')
    schedule_exists = self.model.schedule_exists(date_selected, schedule_selected)
    occupied_seats = ''
    if schedule_exists:
        data = self.model.get_occupied_seats(movie_selected, date_selected, schedule_selected)
        occupied_seats = data[1].split(',')
    else:
        occupied_seats = ['ZZZ',]
    seats = []
    hall_capacity = self.model.get_hall_capacity(movie_selected, schedule_selected.strip())
    for x in range(hall_capacity[0]):
        for y in range(hall_capacity[1]):
            seats.append(f'{string.ascii_lowercase[x].upper()}{y+1}')
    available_seats = [b for b in seats if
        all(a not in b for a in occupied_seats)]
    self.view.msg(', '.join(available_seats))
    seat_selected = '0'
    while seat_selected.upper() not in available_seats:
        seat_selected = input().upper()
        if seat_selected == '0':
            seat_selected = input().upper()
        elif seat_selected.upper() not in available_seats:
            self.view.error('El asiento no está disponible, escoge otro:')
            self.view.msg(', '.join(available_seats))

    if not schedule_exists:
        movie_schedule_id = hall_capacity[2]
        self.model.create_function_schedule(movie_schedule_id, date_selected)

    function_schedule_id = self.model.get_function_schedule_id(movie_selected, date_selected, schedule_selected.strip())
    self.model.create_ticket(function_schedule_id, self.user_id, seat_selected)
    self.view.show_order_details(movie_selected, date_selected, schedule_selected, seat_selected)
    self.view.msg('¡Compraste el boleto de manera exitosa!, presiona cualquier tecla para continuar.')
    tmp = input()
    self.main_menu(self.is_admin)
```

Consulta SQL para obtener los boletos comprados por un cliente:

```
-- Get user tickets
SELECT movies.name, function_schedules.date, movie_schedules.time, halls.name, GROUP_CONCAT(tickets.seat SEPARATOR ",") AS seats FROM user_tickets
JOIN tickets ON user_tickets.user_id = tickets.user_id
JOIN function_schedules ON tickets.function_schedule_id = function_schedules.function_schedule_id
JOIN movie_schedules ON function_schedules.movie_schedule_id = movie_schedules.movie_schedule_id
JOIN movies ON movie_schedules.movie_id = movies.movie_id
JOIN halls ON movie_schedules.hall_id = halls.hall_id
AND user_tickets.user_id = USERID GROUP BY function_schedules.function_schedule_id ORDER BY function_schedules.date
```