UNIVERSIDAD DE GUANAJUATO

División de Ingenierías Campus Irapuato-Salamanca

Licenciatura en Ingeniería en Sistemas Computacionales

Asignatura: Sistemas de información.

Práctica intermedia: Sistema de compra de boletos de cine.

Fecha de entrega: 29 de mayo del 2020.

Doctor: Dr. Juan Carlos Gomez Carranza.

Alumnos: Vicente Ramírez González

NUA: 145714

Salamanca, Guanajuato a 20 de mayo del 2020

Introducción

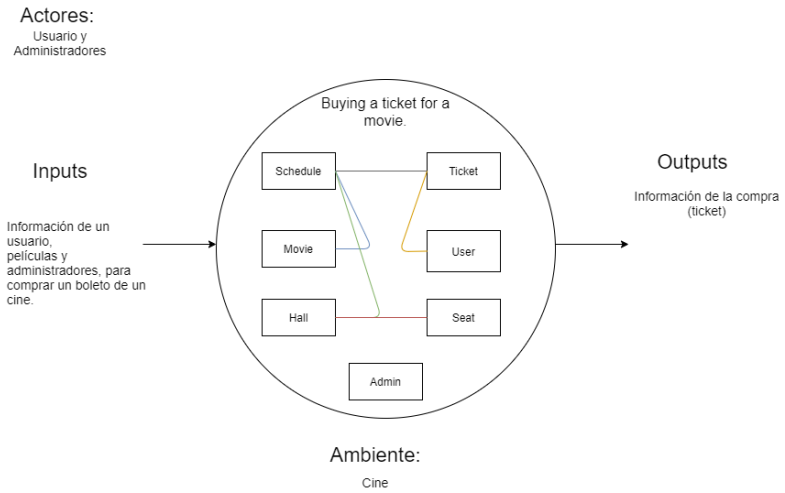
Un sistema de información es un conjunto de datos que interactúan entre sí con un fin común. En informática, los sistemas de información ayudan a administrar, recolectar, recuperar, procesar, almacenar y distribuir información relevante para los procesos fundamentales y las particularidades de cada organización. La importancia de un sistema de información radica en la eficiencia en la correlación de una gran cantidad de datos ingresados a través de procesos diseñados para cada área con el objetivo de producir información válida para la posterior toma de decisiones.

La práctica consiste en la elaboración del diseño y la implementación de un sistema para una compra de boletos en un cine como Cinepolis o Cinemex. En donde se el usuario y el administrador tendrán distintas funciones.

El sistema deberá ser implantando, usando la modelo vista controlador el cual es uno de los patrones arquitectónicos más utilizados desde su facilidad de explicación e implementación. El modelo consiste en una lógica de aplicación para que interactúe con la base de datos. Incluye todas las definiciones de datos (tablas) y procesos (CRUD) para administrar la información de la base de datos. La vista es todo con lo que los usuarios interactúan (interfaz de usuario o front-end). Incluye los archivos para definir el estilo de cómo presentar los elementos visuales en la pantalla (HTML, CSS, JavaScript, etc.). elementos visuales en la pantalla (HTML, CSS, JavaScript, etc.). Y el controlador actúa como intermediario entre la vista y el modelo. Recibe las solicitudes del usuario que se realizan en la vista y las comunica al modelo para realizar un proceso (CRUD, manipulación, etc.).

Metodología

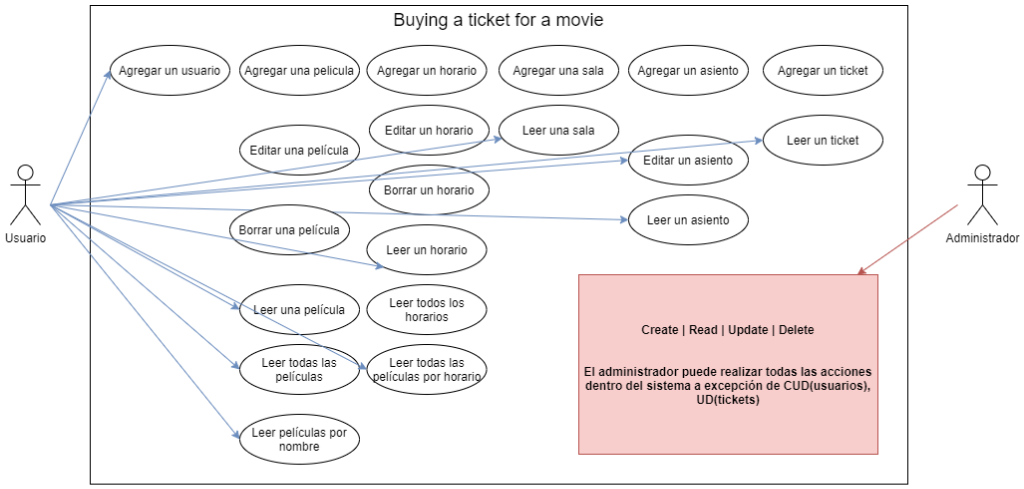
Lo primero que se realizo en esta práctica fue el diseño modular para nuestro sistema, el cual se muestra en la siguiente imagen.



En la imagen se puede apreciar el nombre del sistema, así como sus componentes(módulos) que lo conforman, así también tiene las salidas y entradas que se consideraron para este sistema. Adicionalmente se puede observar los actores y el ambiente que interactúa con el sistema.

En los componentes, se decidió de la siguiente manera: Un horario tendrá relación con el boleto(ticket), para así poder desplegar información relevante del horario dentro del boleto(ticket), una película (movie) esta relacionada con un horario, para saber el nombre de la película que se transmitirá en ese horario, para una sala también tendrá relación con los horarios, para saber a que sala pertenece el horario, para una sala (hall), esta relacionada a un asiento. El usuario(user) tiene relación con el ticket, de igual manera para desplegar información relevante y por último el administrador (Admin) no tiene relación alguna con ningún modulo.

El siguiente paso fue la realización del diseño de casos de uso del sistema, el cual se aprecia en la siguiente imagen:



Nombre:
- Borrar una película.

Suposiciones:
- Sea administrador.

Precondiciones:
- Que la película exista.
- Que se halla seleccionado la opción de borrar una película.
- Que halla confirmado la acción

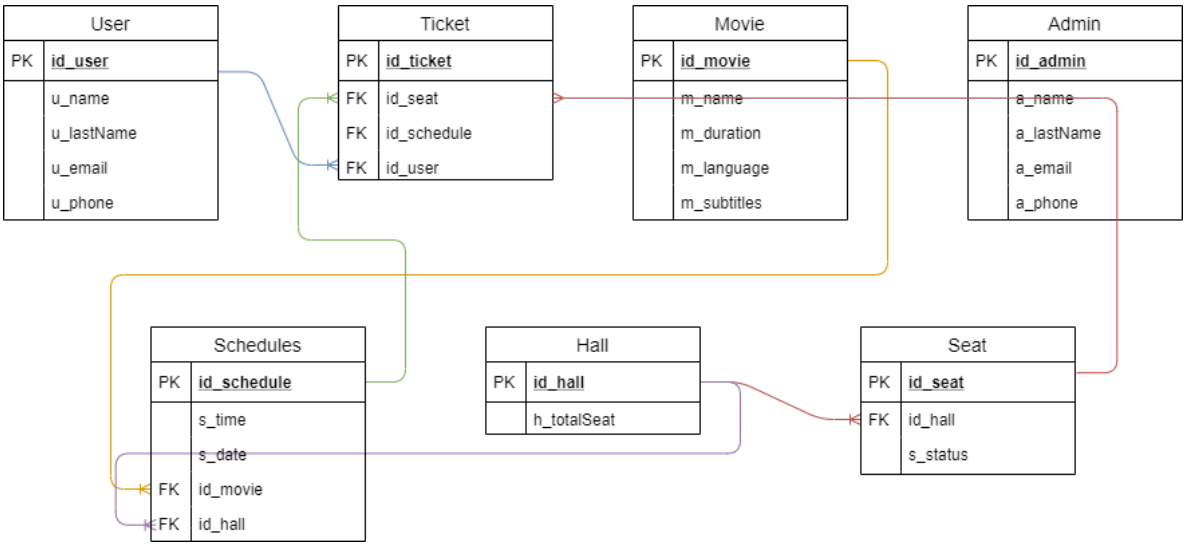
Inicio:
- Click sobre eliminar película

Flujo de eventos:
- S. Muestra un menú sobre eliminar la película.
- A. Ingresa el id de la película que se desea eliminar.
- S. Verifica que exista la película ingresada.
- S. Muestra un mensaje de confirmación para poder eliminar la película.
- A. Acepta la confirmación de la acción.
- S. Muestra un mensaje confirmando que la película se borro.

Postcondición:
- Se actualiza la base de datos.

En el cual se puede observar el nombre del sistema, así como las operaciones que se consideraron más relevantes, para el sistema, además los actores que interactuaran con las operaciones del sistema, además el caso que se decidió explicar fue el de “borrar una película”.

Hasta este punto se empezó con el diseño relacional de la base de datos, el cual se muestra en la siguiente imagen:

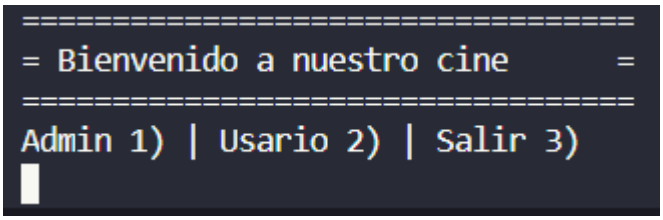


Como se observa el diseño contiene las tablas de nuestra base de datos, la cual esta conformada por 7 tablas, las cuales son User, Ticket, Movie, Admin, Schedules, Hall y Seat. La estructura de estas es muy similar, se comenzará explicando la tabla User, esta contiene 5 campos, en donde su llave primaria (PK) es el id del usuario, la tabla Ticket contiene 4 campos, en donde la llave primaria es el id_ticket y los otros tres campos son llaves foráneas (FK) como el id_seat, id_schedule y id_user. El id_seat es una llave foránea que hace referencia a la tabla Seat, la cual contiene 3 campos, donde la llave primaria es id_seat, y una llave foránea id_hall, s_status, indicando si el asiento esta disponible o ocupado. La siguiente llave foránea de Ticket es id_schedule, la cual esta haciendo referencia a la tabla Schedules, la cual contiene 5 campos, los cuales son id_schedule la cual es la llave primaria (PK), s_time que indica la hora de la función, s_date, indicando la fecha de la función, id_movie, la cual es una llave foránea que hace referencia a la tabla Movie, la cual contiene 4 campos, los cuales son, id_movie como llave primaria, m_name, d_duration, m_language y m_subtitles. Y el último campo de Schedules es id_hall, la cual es una llave foránea que hace referencia a la tabla Hall, esta coniene 2 campos, id_hall como llave primaria y h_totalSeat, indicando el número total de asientos que hay dentro de esa sala._Por ultimo la tabla Admin, la cual contiene 5 campos, id_admin la llave primaria, a_name, a_lastName, a_email y a_phone, esta tabla no tiene relación con ninguna otra, ya que se consideró que, para el funcionamiento de todas estas, no se requiere la información de un

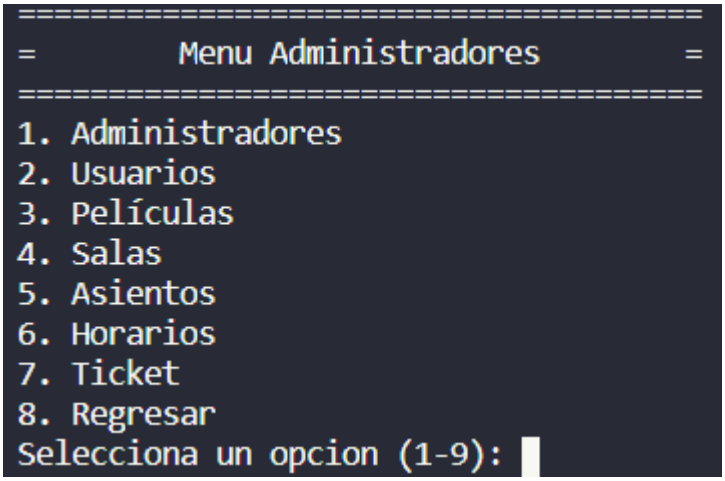
administrador. Y por último hay que mencionar que el diseño relacional cumple con las 3 normas formales, no se tuvo que aplicar ninguna ya que, al momento de la creación del diseño, se definieron bien los puntos para poder realizarla sin problemas.

A partir de este punto y una vez generado el código SQL para la implementación de nuestra base de datos, que corresponde al diseño relacional visto previamente, si inicio con la implementación del sistema.

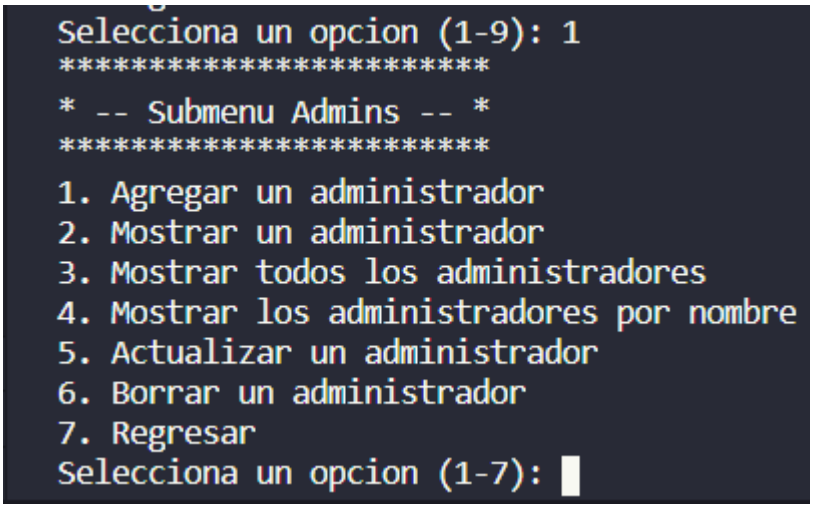
El sistema se comporta de la siguiente manera, al momento que el sistema es iniciado, este despliega un menú, en donde podrás seleccionar si eres administrador o eres usuario, y si no es así una tercera opción para poder salir del sistema.



El sistema tiene dos tipos de menú, para los administradores y para los usuarios, si tu respuesta es 1, despliega el siguiente menú.



En la imagen se puede apreciar todas las operaciones relacionadas a los distintos campos que un administrador puede realizar, una vez ingresada una opción válida, se nos desplegarán submenús para las acciones permitidas a un administrador.



Hasta este punto se puede apreciar como las funciones del sistema, en conjunto a las vistas, se despliegan en el sistema.

Ahora hablaremos del modelo, bien el modelo almacena todas las funciones que se usan para interactuar con las querys de MySQL para poder administrar la información que entra a nuestro sistema. En donde las querys cuentan con segementos como SELECT's UPDATE´S, DELETE´S e INSERT´s, estos según la necesidad de cada función para una mejor experiencia.

Una nota importante es que, al momento del desarrollo de la app, se creaba primero la vista que tendría el submenú, después se hacia la función que contendría el query para el modelo y por último se realizaba la función del controlador, para poder desplegar de manera correcta los datos y se testeaban para ver si había errores y en caso de no haber se pasaba a la siguiente función.

Discusión

Realizar esta práctica permitió, experimentar como se trabaja usando una arquitectura MVC, también se pudieron observar más a fondo los elementos para hacer un buen diseño del sistema, si bien al sistema le hace ser más dinámico, así como incluir validaciones y en su parte visual poder mejorarla con un Framework, se creé que se pudieron aplicar y demostrar los conocimientos previos de las clases, Los diseños que se muestran en este documento son los finales, pero en el transcurso de desarrollo se cambiaran un par de veces estos diseños para llegar a lo que son ahora, teniendo como fin que tuvieran relación con nuestro sistema y nuestra base de datos.