Sistema de Reserva de Boletos para Cinemax

Diseño Y Programación Orientada a Objetos

Carlos Esteban Ramirez Correa

202310150

25/09/2024

1. **Introducción**

   El documento presenta un análisis del sistema de reserva de entradas de la cadena de cines CineMax. El objetivo de este sistema es facilitar la proyección de películas y la gestión de la reserva de entradas, garantizando la comodidad de clientes y empleados. El análisis va acompañado de un modelo UML que describe la estructura del sistema y sus principales componentes.

2. **Descripción de las Clases**

   - **Cine:**
     - **Atributos:**
       - nombre: String, nombre del cine.
       - dirección: String, dirección del cine.
     - **Métodos:**
       - agregarSala(Sala sala): void. Agrega una sala al cine.
       - listaDeFunciones(): List<Funcion>. Programa una función en una sala especifica.
   - **Sala:**
     - **Atributos:**
       - numeroSala: Int. Identificador de la sala.
       - capacidadAsientos: Int. Capacidad máxima de asientos.
       - tipoTecnologia: String. Tipo de tecnología de la sala.
     - **Métodos:**
       - calcularCapacidadDisponible(): Int. Calcula los asientos disponibles en la sala.
       - listarFunciones(): List<Funcion>. Programa una función en una sala especifica.

   - **Asiento:**
     - **Atributos:**
       - numero: Int. Número del asiento
       - numeroFila: Int. Numero de la fila donde está el asiento.
       - tipoAsiento: String. El tipo de asiento, si es regular o premium.
       - estado: boolean. Si esta libre es true, si está reservado es false.
     - **Métodos:**
       - marcarReservado(boolean estado): void. Cambia el estado de true a false
       - liberar(boolean estado): void. Cambie el estado del asiento de false a true.
   - **Función:**
     - **Atributos:**

        - fecha: Date. Fecha de la función.
        - hora: Time. Hora de la función.
    - **Métodos:**
        - verAsientosDisponibles(Asiento asiento): boolean. Visualiza los asientos disponibles que hay en la sala.
        - cancelarReserva(Asiento asiento): void. Cancela la reserva de un asiento de la función.
- **Películas:**
    - **Atributos:**
        - título: String. Es el título de la película.
        - género: String. Es el género de la película.
        - duración: Int. Es la duración de la película.
        - clasificación: String. La clasificación de la película.
        - fechaDeEstreno: Date. La fecha de estreno de la película.
    - **Métodos:**
        - mostrarInfo(): String. Muestra la información de una película.
        - listaDeFuncionesDisponibles(): List<Funcion>. Muestra la lista de funciones disponibles de la película.
- **Cliente:**
    - **Atributos:**
        - id: Int. Id unico del cliente
        - correoElectronico: String. Correo del cliente
        - contrasenia: String. Contraseña del cliente.
        - historialCompras: List<Pago>. Historial de las compras realizadas por el cliente.
    - **Métodos:**
        - verHistorial(): List<Pago>. Muestra la lista de pagos que ha realizado el cliente.
        - acumularPuntos(Pago pago): void. Acumula puntos de acuerdo con la naturaleza del pago.
        - explorarPeliculas(): List<Peliculas>. Muestra la lista de películas que hay disponibles ene se momento.
        - seleccionarFuncion(Pelicula pelicula): List<Funcion>. Selecciona la función de acuerdo con la película.
        - reservarAsientos(Funcion funcion, List<Asiento> asientos, String tipoPago, Pago pago): void. Realiza la reserva de un conjunto de asientos para una función específica.
- **ProgramaLealtad:**
    - **Atributos:**
        - puntosAcumulados: Int. Puntos acumulados por el cliente.
        - nivelMembresia: String. Nivel de membresía que tenga el cliente.
        - descuentosDisponibles: List<String>. Lista con los descuentos y beneficios a los que tiene acceso el cliente por su membresía.

        - **Métodos:**
            - verNivel(): Int. Muestra el nivel de la membresía
            - consultarDescuentos(): List<String>. Muestra la lista de descuentos disponibles que tiene el cliente.
    - **Pago:**
        - **Atributos:**
            - tipoPago: String. El tipo de tarjeta que se usa, si debito o crédito.
            - monto: Double. Precio de la transacción.
            - fechaTransaccion: Date. Fecha en la que se hizo el pago.
            - sillasCompradas: List<Asientos>. Lista de asientos comprados.
        - **Métodos:**
            - calcularMonto(): Double. Calcula el valor a pagar.
    - **Empleados:**
        - **Atributos:**
            - correoElectronico: String. Correo electrónico del empleado.
            - contrasenia: String. Contraseña de acceso del empleado.

        - **Métodos:**
            - configurarFuncion(Funcion funcion): void. Programa una nueva función en el sistema.
            - actualizarInfoPelicula(Pelicula pelicula): void. Gestiona las películas disponibles en el cine.

3. **Relaciones entre clases:**

    - **Película - Función:** Una película puede tener varias funciones y cada función pertenece a una sola película.
    - **Cine – Sala:** Un cine puede tener varias salas y cada sala pertenece a un solo cine.
    - **Sala – Asiento:** Una sala contiene varios asientos y cada asiento pertenece a una sola sala.
    - **Sala – Función:** Una función pertenece a una sola sala y una sala puede tener varias funciones
    - **Cliente – Pagos:** Un cliente puede realizar varios pagos y cada pago pertenece a un cliente.
    - **Programa – Cliente:** Un programa pertenece a un cliente y un cliente pertenece a un cliente.
    - **Cliente – Cine:** Un cine puede tener varios clientes y cada cliente pertenece a un cine.

4. **Reglas del dominio:**

    - Un cliente no puede reservar asientos que ya están ocupados.
    - Un cliente solo puede acumular puntos si pertenece al programa lealtad.
    - Un cliente debe proporcionar un método de pago valido para completar la reserva.
    - Un empleado solo puede programar funciones en salas disponibles.
    - Cada función debe tener una sala y una película.
    - Los asientos premium tienen un costo adicional.
    - Un asiento pertenece a una única sala.
    - Un pago esta asociado solo a una función y a un cliente.

5. **Conclusiones**

    Este sistema propone una solución robusta para las reservas y las funciones de los cines de CineMax. Considerando varios casos, las clases y relaciones modeladas permiten gestionar de manera eficiente la información de clientes, películas, funciones y pagos.

6. **Diagrama UML**

    El diagrama UML fue creado con el programa UMLET y esta adjunto en el repositorio.