



## Cine SpiderDAM: TAD Revenge

Ya conocemos nuestro cine. Lo que no sabíamos es que regresaría para ayudarnos a avanzar un poco como programadores.

Nuestro **cine**, tiene una sola **sala**, la cual tiene un número determinado de **butacas** organizadas en **filas** y **columnas**.

Por otro lado, debemos saber que tenemos butacas normales y butacas VIP, y butacas de ofertas. Una butaca tiene un identificador único, una fecha de creación, su fila, columna, su estado: LIBRE, OCUPADO y RESERVADO. Cada una de ellas deberá tener un precio que debemos iniciar al comenzar el sistema.

El sistema tiene un **administrador** que es un usuario especial, por lo que al comenzar el sistema se debe pedir el nombre de usuario del administrador, que siempre ser tres letras de la a a la z en minúsculas seguida de tres dígitos: por ejemplo ad123, jod333. Además necesitaremos los datos de cualquier **usuario** que son el nombre real: Pepito de los Palotes, y su correo electrónico.

Seguidamente iniciaremos la configuración de la sala, indicando el número de filas (entre 4 y 10) y de columnas (5 y 15). Seguido del número de butacas con ofertas: (1 a 5) y VIP (3-10). El resto será normales. Seguiremos con el precio de las butacas en oferta (5,00 a 7,00), normales (7,00 a 10,00) y VIP (10,00 a 15,00). Con ellos crearemos un objeto de configuración que nos servirá para instanciar la sala, teniendo en cuenta que su instancia debe ser única.

Debemos tener en cuenta que si la butaca es VIP lleva añadido un complemento que puede elegir el usuario a la hora de comprar/reservar la entrada la entrada: Palomitas (3€), Bebida (5€), Completo (7€).

Debemos tener en cuenta que as filas siempre se referenciarán con una letra, por lo que deberás comenzar con la letra A, seguida de de tantas letras como filas haya. Por ejemplo si hay 5 tendremos A, B, C, D, E. Te recomiendo que aprendas a sumar characters.

A partir de aquí se nos presentará un menú donde podremos:

- Ver estado de la sala. Imprimirá el estado de la sala. Teniendo en cuenta que usaremos colores para diferenciar las butacas libres, ocupadas reservadas, además las que son VIP, normales y en Oferta. Puedes hacerlo usando colores de fondo y de letra o fondo y emoticonos. Se debe mostrar el número de la columna y la letra de la fila.

- Reservar una butaca. Dada una coordenada por ejemplo B3, reservará la butaca correspondiente a la fila dos y columna 3. Se mostrará un mensaje si ha sido posible o no, y la información de la butaca. Cada vez que reservemos una entrada, debemos almacenarlas en una repositorio, siempre ordenada por filas y si es posible en caso de empate con columnas. Una reserva debe tener los datos de la reserva como son identificador, butaca y momento. La reserva se asociará a un **cliente**, que es un usuario normal, con un identificador propio o número de cliente.

- Anular reserva, el contrario al anterior si existe una reserva partiendo de las coordenadas de la butaca (B3) de la reserva, se buscará en su repositorio para obtener los datos de la butaca y con ellos anular la reserva volviendo la butaca a su estado anterior a la misma.

- Comprar entrada, se le pedirá los datos del usuario y la coordenada de la butaca (C4), pasando a estar ocupada si es posible. Si la butaca esta reservada y los datos del usuario de la reserva coinciden, dicha reserva pasará a estar confirmada y la butaca ocupada.

Por cada entrada comprada generaremos un **Ticket**, el cual tiene los datos de la compra como son identificador de la misma, butaca, tipo, precio, momento y el nombre del administrador que lo ha atendido. Estos tickets se ordenarán por defecto en un repositorio de más nuevo a más antiguo. Si la butaca está reservada. Debemos tener en cuenta que el precio del ticket si la butaca es VIP será verificado incrementado por el complemento que elija.

- Devolver entrada, se le pedirá el identificador del Ticket, y si existe, anulará el ticket y devolverá la butaca al estado anterior a la venta.

- Ver reservas. Mostrará la lista de reservas siguiendo el orden indicado.

- Ver ventas. Mostrará las ventas de Tickets según el orden indicado.

- Recaudación. Muestra el importe total de Tickets vendidos.

- Informe: Muestra información del estado de la sala en base a: número de reservas, butacas libres, butacas vendidas. Porcentaje de ocupación. Recaudación. Número de butacas en oferta libres, ocupadas o reservadas. Número de butacas normales libres, ocupadas y reservadas y número de butacas vip libres, ocupadas y reservadas.

### **Te en cuenta:**

Haz un diagrama de clases antes de implementar y un diagrama de secuencia de sistema.

Todas las estructuras de datos deben ser dinámicas y estar preparadas para ser usadas con cualquier dato, es decir Genéricas, definiendo el TDA correspondiente.

Sala debe cumplir con una instancia única que se inicia parametrizada con un objeto de configuración.

Se debe implementar los comportamientos en base a interfaces.

Todas las entrada y filtros de datos deben usar expresiones regulares para ver si son correctas.

Todas las salidas que incluyan decimales deben estar limitadas a 2 decimales.

Aunque en este problema no es necesario, dentro del menú o bucle que gestiona la lógica de opciones, debes implementar el double buffer para asegurar que lecturas y escrituras en la estructura no interfieren entre ellas generando errores o inconsistencias en su funcionamiento. Repito en este problema no tiene sentido, pero es bueno que lo practiques.

Cualquier cosa que supongas debes analizarla con cuidado y justificar su uso.