

Enunciado: Gestión de un Cine (POO en Java)

Diseña una aplicación de consola en Java que gestione la venta de entradas en un cine usando Programación Orientada a Objetos. Implementa las clases **Pelicula**, **SalaDeCine**, **Cine** y **Usuario** siguiendo estas indicaciones:

Modelado (mínimo)

- **Pelicula**: título, duración (min), clasificación por edad (opcional), precio (opcional), id/código.
- **SalaDeCine**: código/nombre, capacidad (entera), **pelicula** proyectada, **butacas** (array de Usuario del tamaño de la capacidad; null indica asiento libre).
- **Cine**: nombre, **salas** (array de SalaDeCine).
- **Usuario**: nombre y/o DNI/ID. (Puedes añadir edad/saldo si lo ves útil, pero no es obligatorio.)

Reglas de negocio

1. El **Cine** contiene un array de salas (por ejemplo, 3–5).
2. Cada **SalaDeCine** proyecta **una** película y dispone de un array de butacas (asientos) donde cada posición puede estar ocupada por un Usuario o null si está libre.
3. La aplicación trabajará con **un array de 3 usuarios** (p. ej., Usuario[] compradores = new Usuario[3]), simulando tres clientes que comprarán entradas de forma secuencial.
4. A cada **Usuario** se le preguntará **cuántas entradas** desea ($n \geq 1$).
5. La aplicación mostrará **solo las películas** (y sus salas correspondientes) que **tengan suficientes asientos libres** para cubrir la cantidad solicitada.
6. Tras elegir película/sala, el sistema **asignará automáticamente las primeras butacas disponibles** (las de menor índice en el array) y registrará al Usuario en dichas posiciones.
7. Debe mostrarse por pantalla el **resumen de la compra**: película, sala, cantidad y **números de butaca asignados**.
8. Si no hay salas con asientos suficientes para la cantidad solicitada, se informará al usuario y no se efectuará la compra.

Flujo del programa (menú)

Implementa un **menú de consola** que permita, al menos:

1. **Listar películas y disponibilidad:** para cada sala, mostrar película proyectada y número de asientos libres/ocupados.
2. **Comprar entradas:**
 - Pedir cantidad de entradas al usuario activo.
 - Mostrar únicamente las opciones (película/sala) con asientos suficientes.
 - Permitir seleccionar una de las opciones.
 - Asignar las primeras butacas libres y mostrar el detalle.
3. **Mostrar ocupación por sala:** imprime el mapa/índices de butacas, indicando cuáles están ocupadas y por quién.
4. **Salir.**

El menú debe ejecutarse **para cada usuario del array de 3 usuarios**, uno tras otro (p. ej., “Turno de Usuario 1...”, luego Usuario 2, etc.). Tras completar los tres turnos, la aplicación puede finalizar.

Requisitos técnicos

- Encapsula atributos y proporciona constructores y métodos de acceso adecuados.
- Métodos sugeridos:
 - En SalaDeCine: int asientosLibres(), boolean hayEspacio(int n), int[] asignarButacas(Usuario u, int n) (devuelve los índices asignados o un array vacío si no procede).
 - En Cine: métodos para obtener las salas con disponibilidad $\geq n$ y para listar estado.
- Valida entradas del usuario (cantidades positivas, opciones válidas, etc.).
- Mensajes claros de éxito/error y trazabilidad de las operaciones.

Entrega esperada: código fuente organizado por clases, ejecutable por consola, que cumpla las reglas y el flujo indicados.