

Materia:	Programación II		
Nivel:	2º Cuatrimestre		
Tipo de Examen:	Segundo Parcial		
Apellido ⁽¹⁾ :		Fecha:	Semana de Julio
Nombre/s ⁽¹⁾ :		Docente a cargo ⁽²⁾ :	Benitez
División ⁽¹⁾ :	124-321-325	Nota ⁽²⁾ :	
DNI ⁽¹⁾ :		Firma ⁽²⁾ :	

(1) Campos a completar solo por el estudiante en caso de imprimir este enunciado en papel.

(2) Campos a completar solo por el docente en caso de imprimir este enunciado en papel.

Desarrollar un sistema de venta de entradas para un cine que permita a los usuarios:

- Registrarse o iniciar sesión como cliente.
- Ver las salas disponibles y su estado (butacas libres y ocupadas).
- Comprar una entrada para una función seleccionada.
- Persistir el estado del sistema (salas, butacas ocupadas y clientes) mediante serialización.

Clases principales

Dominio:

Cliente: nombre, email, contraseña.

Butaca: número, fila, estado (ocupada/libre).

Sala: número, película, matriz de butacas.

Entrada: cliente, sala, butaca.

Cine: contiene lista de salas y entradas.

Persistencia:

PersistenciaDatos: métodos para guardar y cargar los objetos con serialización.

Interfaz JavaFX

Pantalla de Login/Registro.

Pantalla de selección de sala y película.

Visualización de butacas (ocupadas/libres).

Confirmación de compra.

Opcional: resumen de entradas compradas.

Persistencia

Al cerrar la aplicación, se guarda el estado del cine (cine.ser).

Al iniciar, se recupera el estado si existe.

Estructura de paquetes sugerida

cine/

- | modelo/ → Cliente, Sala, Butaca, Entrada, Cine
- | vista/ → Ventanas JavaFX (LoginView, SalaView, etc.)
- | controlador/ → Lógica de login, compra, navegación
- | persistencia/ → PersistenciaDatos.java

Pasos del desarrollo

Modelar las clases del dominio.

Diseñar la persistencia de estado del sistema.

Implementar la lógica de negocio (compra, login).

Desarrollar la GUI con JavaFX.

Integrar todo usando MVC.

Criterios de corrección:

1. Modelado del dominio y uso de POO (20 puntos)

Criterio	Puntos
Clases correctamente diseñadas con atributos adecuados (Cliente, Butaca, Sala, etc.)	5
Uso correcto de encapsulamiento (getters/setters, acceso privado)	5
Relación entre clases correctamente modelada (Sala → Butacas, Entrada → Cliente/Sala/Butaca)	5
Buen uso de colecciones (List, Array[][], etc.)	5

2. Persistencia (15 puntos)

Criterio	Puntos
Implementación de serialización (ObjectOutputStream / ObjectInputStream)	5
Archivo cine.ser generado correctamente	5
El sistema recupera correctamente el estado al iniciar	5

3. Lógica de negocio y controlador (15 puntos)

Criterio	Puntos
Registro y login de cliente funcional	5
Compra de entradas válida, verifica disponibilidad de butaca	5
Uso de controlador o separación de responsabilidades	5

4. Interfaz gráfica con JavaFX (25 puntos)

Criterio	Puntos
Pantalla de Login y Registro funcional y clara	5
Visualización de salas y películas correctamente implementada	5

Criterio	Puntos
Estado de las butacas (libres/ocupadas) visible y seleccionable	5
Flujo de interacción coherente y usabilidad básica	5
Uso correcto de Scene, VBox, Alert, etc.	5

5. Estructura del proyecto y MVC (10 puntos)

Criterio	Puntos
Paquetes organizados correctamente (modelo, vista, controlador, persistencia)	5
Separación entre lógica y vista (uso de controlador)	5

6. Pruebas y funcionamiento general (10 puntos)

Criterio	Puntos
El sistema funciona de punta a punta (registro → selección → compra → guardar)	5
Manejo básico de errores: login inválido, butaca ocupada, butaca fuera de rango, etc.	5

7. Código limpio y buenas prácticas (5 puntos)

Criterio	Puntos
Nombres de clases, variables y métodos claros y significativos	2
Comentarios relevantes donde sea necesario	1
Ausencia de código duplicado o sin usar	2

Resultado Final

Categoría	Puntaje
Modelado de clases	/20
Persistencia	/15
Lógica de negocio	/15
Interfaz JavaFX	/25
Estructura y MVC	/10
Funcionamiento general	/10
Código y buenas prácticas	/5

TOTAL