

## IF0004 – Desarrollo de Software II

### Proyecto del curso

**VALOR DEL PROYECTO:** 20%

**FECHA DE INICIO DEL PROYECTO:** 30 de enero del 2026

**FECHA DE ENTREGA FINAL:** 26 de febrero del 2026 – 1:00 PM

### I. REQUERIMIENTOS GENERALES

El auditorio de la Sede Regional desea modernizar su sistema de venta de entradas para conciertos y obras de teatro. Actualmente, todo se lleva en papel. Se le ha contratado a su equipo de desarrollo para crear una aplicación de escritorio robusta, visual y persistente que permita gestionar la cartelera de eventos y la venta de boletos con selección de asientos en tiempo real. El software debe cumplir estrictamente con los siguientes lineamientos técnicos:

- Arquitectura MVC: El código debe estar separado en paquetes model, view y controller.
- El mapa de asientos del auditorio debe representarse obligatoriamente mediante una Matriz (Arreglo Bidimensional)
- Debe usar algoritmos de Búsqueda para encontrar clientes o reservas por ID
- Debe existir una clase padre abstracta Entrada o Boleto.
- Debe implementar Herencia y Polimorfismo con al menos 3 tipos de entradas: EntradaGeneral, EntradaVIP (incluye acceso a lounge) y EntradaEstudiante (aplica descuento)
- Uso de una Interfaz (ej. IVendible o IImprimible)
- La interfaz debe ser gráfica (Swing o JavaFX).
- El mapa de asientos debe ser visual (botones o cuadros): Verde (Libre), Rojo (Ocupado).
- Manejo de Eventos al hacer clic en un asiento.
- Manejo de Excepciones: El sistema no debe "caerse" si el usuario ingresa letras en campos numéricos.
- Archivos de Texto: Al cerrar el programa, la información de las ventas y el estado de los asientos debe guardarse en archivos .txt o .csv. Al abrirlo de nuevo, debe cargar el estado anterior (Persistencia)

### II. MÓDULOS DEL SISTEMA

#### Módulo Administrativo (Back-Office)

- Gestión de Eventos: Crear, editar y eliminar eventos (Nombre, Fecha, Precio Base).
- Reportes: Ver cuánto dinero se ha recaudado por evento (recorrer las estructuras).
- Reinicio: Capacidad de "limpiar" la sala para un nuevo evento (reiniciar la matriz).

#### Módulo de Venta (Front-Office)

- Selección de Asientos: El usuario ve una cuadrícula (Matriz) de, por ejemplo, 10 x 10 asientos.



- Compra:
  - El usuario selecciona un asiento disponible.
  - El sistema pregunta el tipo de boleto (VIP, General, Estudiante).
  - El sistema calcula el precio final usando Polimorfismo (ej. VIP suma costo, Estudiante resta porcentaje).
  - El sistema marca el asiento como ocupado en la matriz y actualiza la vista.
- Facturación: Generar un pequeño "ticket" en archivo de texto con el detalle de la compra.

### III. ENTREGABLES

- Código Fuente: Proyecto completo (NetBeans/IntelliJ/Eclipse) compilable.
- Manual Técnico:
  - Diagrama de Clases UML actualizado.
  - Explicación de cómo se implementó el patrón MVC.
- Ejecutable: Archivo .jar funcional.

### IV. RÚBRICA

Criterio	Puntos	Detalle
Arquitectura MVC y Diagrama de clases	15 pts	Separación correcta de modelo, vista y controlador.
Buenas Prácticas	10 pts	Código limpio, nombres de variables correctos, indentación.
POO (Herencia/Poli)	20 pts	Uso correcto de clases abstractas, interfaces y sobrescritura.
Manejo de Matriz	15 pts	Lógica correcta para reservar/liberar espacios en la matriz.



<b>Interfaz Gráfica</b>	15 pts	Usabilidad, diseño y manejo de eventos de botones.
<b>Persistencia (Archivos)</b>	15 pts	Los datos se guardan y recuperan correctamente tras cerrar la App.
<b>Funcionamiento General</b>	10 pts	El programa cumple con los requerimientos de negocio y no tiene errores fatales.