



## Laboratorio 2. Arreglos y excepciones - Análisis y Diseño

### Análisis

#### Requisitos funcionales:

- El sistema debe de reservar asientos.
- El sistema debe de mostrar la disponibilidad de asientos.
- El sistema debe de cancelar reservas.
- El sistema debe de calcular los ingresos por sección.

#### Clases:

- Secciones: Su propósito es almacenar los ArrayList de las distintas secciones.
- Gestion: Su propósito es realizar las distintas tareas necesarias.
- DriverProgram: Su propósito es interactuar con el usuario.
- GestionGUI: Clase que proporciona una interfaz gráfica para la gestión de reservas de asientos en el auditorio.

#### Propósito de atributos:

##### Secciones:

- vip: int[][]: Su propósito es almacenar la disponibilidad de los asientos de las diferentes filas de la sección VIP. 0 no disponible, 1 disponible.
- platino: int[][]: Su propósito es almacenar la disponibilidad de los asientos de las diferentes filas de la sección Platino. 0 no disponible, 1 disponible.
- oro: int[][]: Su propósito es almacenar la disponibilidad de los asientos de las diferentes filas de la sección Oro. 0 no disponible, 1 disponible.
- plata: int[][]: Su propósito es almacenar la disponibilidad de los asientos de las diferentes filas de la sección Plata. 0 no disponible, 1 disponible.
- Archivo: String: El nombre del archivo CSV.
- Sección: int[][]: La matriz de asientos a cargar.

##### Gestion:

- secciones: Secciones: Su propósito es crear un nuevo objeto secciones, para poder realizar las distintas tareas necesarias.
- Sección: int: Número de la sección (1 para VIP, 2 para Platino, 3 para Oro, 4 para Plata).
- Fila: int: Número de la fila del asiento.
- Asiento: int: Número del asiento en la fila.

DriverProgram:

- gestión: Gestion: Su propósito es crear un nuevo objeto gestión, para poder interactuar con el usuario.
- teclado: Scanner: Su propósito es permitir la entrada de datos por el usuario.

GestionGUI:

- gestión: Gestion: Su propósito es crear un nuevo objeto gestión, para poder interactuar con el usuario.
- textArea: JTextArea: Su propósito es ser visto en la interfaz gráfica.

### **Propósito de métodos:**

Secciones:

- inicializarSecciones(): void. Inicializa todas las secciones de asientos con el valor 1 (disponible).
- getVip(): int[[]]. Devuelve la matriz que representa la sección VIP.
- getPlatino(): int[[]]. Devuelve la matriz que representa la sección Platino.
- getOro(): int[[]]. Devuelve la matriz que representa la sección Oro.
- getPlata(): int[[]]. Devuelve la matriz que representa la sección Plata.
- cargarDatos(): void. Carga los datos de las secciones desde archivos CSV.
- cargarSeccion(String archivo, int[[]] seccion): void. Carga los datos de una sección desde un archivo CSV.
- guardarDatos(): void. Guarda los datos de las secciones en archivos CSV.
- guardarSeccion(String archivo, int[[]] seccion): void. Guarda los datos de una sección en un archivo CSV.

Gestion:

- reservarAsientos(int seccion, int fila, int asiento): String. Su propósito es revisar si el asiento seleccionado esta reservado y si no lo está, reservarlo.
- mostrarDisponibilidad(int seccion): String. Su propósito es mostrar la disponibilidad de la sección seleccionada.
- cancelarReserva(int seccion, int fila, int asiento): String. Su propósito es revisar si el asiento seleccionado esta reservarlo y si lo está, cancelar la reservación.
- calcularIngresos(): String. Su propósito es realizar las operaciones necesarias por los asientos reservados para mostrar los ingresos por sección.

- obtenerListaSeccion(int seccion): int[[]]. Obtiene la matriz de asientos para una sección específica.

DriverProgram:

- Main(args String[]): Su propósito es interactuar con el usuario.

GestionGUI:

- reservarAsiento(): void. Método para reservar un asiento.
- mostrarDisponibilidad(): void. Método para mostrar la disponibilidad de asientos en una sección.
- cancelarReserva(): void. Método para cancelar una reserva de asiento.
- calcularIngresos(): void. Método para calcular los ingresos totales por sección.
- Main(args String[]): Su propósito es interactuar con el usuario.

## **Diseño**

