

Presentación del Proyecto: Sistema de Reservas de Cine

Nombre del Proyecto

Sistema Básico de Gestión de Reservas de Asientos de Cine *Una solución de consola para la administración simplificada de butacas en salas de cine.*

Lógica Utilizada

La lógica central del sistema se basa en una representación sencilla de la sala de cine y en un ciclo de menú interactivo:

- **Representación de la Sala:** La sala de cine se modela como una **matriz (arreglo bidimensional)** de valores booleanos (**`boolean[][]`**).
 - Cada elemento **`true`** en la matriz significa que el asiento está **ocupado**.
 - Cada elemento **`false`** en la matriz significa que el asiento está **disponible**.
 - Esta elección simplifica la lógica de verificación de estado y manipulación.
- **Interacción por Menú:** El programa opera a través de un **menú de opciones** presentado al usuario en la consola. Un **ciclo `do-while`** asegura que el menú se muestre repetidamente hasta que el usuario decida salir.
- **Lógica de Reservas y Cancelaciones:**
 - Para **reservar**, se verifica si el asiento está **`false`** (disponible); si lo está, se cambia a **`true`** (ocupado).
 - Para **cancelar**, se verifica si el asiento está **`true`** (ocupado); si lo está, se cambia a **`false`** (disponible).
- **Cálculo de Ocupación:** Se recorre toda la matriz contando los asientos en estado **`true`** para determinar el número y porcentaje de ocupación y disponibilidad.

Estructuras de Programación Aplicadas

Este proyecto es un excelente ejemplo de aplicación de conceptos fundamentales de programación estructurada en Java:

- **Arreglos Bidimensionales (`boolean[][]`)**
 - **Propósito:** Representar la cuadrícula de la sala de cine (filas y asientos).
 - **Implementación:** Permite acceder a cada asiento de manera directa mediante sus índices `sala[filas][asientos]`.

- **Ciclos (for, do-while)**
 - **for anidados:** Fundamentales para:
 - **Inicializar** todos los asientos como disponibles.
 - **Mostrar** el estado de la sala, recorriendo cada fila y columna.
 - **Contar** asientos ocupados/disponibles para calcular el porcentaje de ocupación.
 - **do-while:** Utilizado para el **bucle principal del menú**, asegurando que el menú se muestre al menos una vez y se repita hasta que el usuario elija salir.
- **Estructuras Condicionales (if, else if, else)**
 - **Validación de entrada:** Comprobar si los números de fila y asiento ingresados por el usuario están dentro de los rangos válidos.
 - **Lógica de negocio:** Determinar si un asiento está disponible para reservar o si está ocupado para cancelar.
- **Estructura de Selección (switch)**
 - **Manejo del menú:** Permite ejecutar bloques de código específicos según la opción que el usuario elija del menú principal (mostrar sala, reservar, cancelar, etc.).
- **Manejo de Excepciones (try-catch)**
 - **Robustez:** Se utiliza para **capturar errores de tipo de dato** (`InputMismatchException`) cuando el usuario ingresa texto en lugar de un número. Esto evita que el programa se detenga inesperadamente y permite solicitar una nueva entrada.

Retos Enfrentados

Durante el desarrollo de este sistema básico, algunos de los retos principales incluyeron:

- **Alineación en la Consola:** Lograr que la representación visual de la sala (números de columna, línea divisoria, filas de asientos) se mostrara perfectamente alineada en la consola, utilizando `System.out.printf()` y calculando los espacios y caracteres de forma precisa. Pequeños desajustes podían romper la cuadrícula.
- **Manejo de Entrada de Usuario:** Implementar un mecanismo robusto (`try-catch` y `scanner.nextLine()`) para que el programa no "estallara" si el usuario introducía un tipo de dato incorrecto (ej. texto en lugar de números para la fila/asiento). Asegurar que el búfer del `Scanner` se limpiara correctamente fue clave.
- **Lógica de Índices (0-based vs 1-based):** Convertir constantemente las entradas del usuario (que son 1-basadas, ej., Fila 1) a los índices 0-basados de la matriz en Java (ej., Fila 0), y viceversa al mostrar al usuario.

Conclusiones

Este proyecto, a pesar de su simplicidad, ha sido una base sólida para comprender y aplicar conceptos fundamentales de programación:

- **Fundamentos Sólidos:** Demuestra la importancia de las estructuras de control (ciclos y condicionales) y las estructuras de datos (arreglos bidimensionales) en la construcción de programas interactivos.
- **Importancia del Manejo de Errores:** Subraya la necesidad de anticipar y manejar errores de usuario para crear aplicaciones robustas y amigables.
- **Diseño de la Interfaz de Usuario (Consola):** Resalta que incluso en una interfaz de consola, la claridad y la alineación visual son cruciales para la usabilidad.
- **Preparación para la Complejidad:** Aunque simple, sienta las bases para futuras expansiones del proyecto, como la adición de múltiples salas, diferentes precios, nombres de clientes, y una interfaz gráfica de usuario.