

# Descripción del Proyecto:

## Sistema de Venta de Boletos para Estadio

### Objetivo del Proyecto:

Desarrollar un sistema en Java que permita gestionar la venta de boletos para un estadio. Los compradores pueden elegir boletos en tres categorías diferentes (VIP, General y Preferencial), cada una con un precio diferente. El sistema incluirá una interfaz gráfica para que los usuarios seleccionen asientos y generará reportes de ventas que se guardarán en archivos de texto.

### Funcionalidades Principales:

#### 1. Interfaz Gráfica de Usuario (GUI):

**10 pts**

- Validar todo el sistema
- El sistema contará con una ventana donde los usuarios podrán:
  - **Seleccionar la categoría de boleto** (VIP, General, Preferencial).
  - **Ver los asientos disponibles** en un formato visual, representado como una matriz de asientos (ocupados y disponibles).
  - **Confirmar la compra** de boletos seleccionando los asientos y realizando el pago.
  - **Ver el precio total** de la compra, que depende de la categoría y los asientos seleccionados.

#### 2. Gestión de Boletos:

**10 pts**

- Los boletos estarán organizados por categoría (VIP, General, Preferencial).

- Cada boleto tiene los siguientes atributos:

- **ID de boleto**
- **Categoría**
- **Precio**
- **Número de asiento o ubicación**
- **Estado (Disponible o Vendido)**

### 3. Generación de Reportes de Ventas:

**10 pts**

- Al completar una compra, el sistema generará un reporte que contendrá:
  - **Fecha y hora de la venta**
  - **Categoría de boleto**
  - **Número de boletos vendidos**
  - **Ingreso total generado**
  - **Detalles de los asientos vendidos**
- Los reportes serán guardados en archivos de texto (.txt), uno por cada día de ventas.

### 4. Validaciones:

**10 pts**

- **Asientos no duplicados:** No se podrá comprar un asiento que ya esté ocupado.
- **Selección de asientos:** Solo podrán seleccionarse los asientos disponibles dentro de la categoría seleccionada.
- **Precios de boletos fijos:** Los precios de las categorías (VIP, General y Preferencial) pueden ser actualizados, pero están definidos en un mapa (HashMap).

## Estructuras de Datos Utilizadas:

A continuación, te detallo las estructuras de datos que se utilizarán para gestionar los boletos, asientos y los reportes de ventas.

### 1. Listas Enlazadas para Boletos: 10 pts

Las **listas enlazadas** se utilizarán para almacenar los boletos de cada categoría, permitiendo una gestión dinámica de los boletos disponibles.

- **Objetivo:** Almacenar los boletos de cada categoría (VIP, General, Preferencial).
- **Cómo se utiliza:**
  - Cada lista enlazada contendrá objetos de tipo **Boleto**, donde cada **Boleto** tendrá los atributos: ID de boleto, categoría, precio, número de asiento y estado (disponible o vendido).
  - **Operaciones:**
    - **Agregar boleto:** Cuando un cliente compra un boleto, se crea un objeto **Boleto** y se inserta en la lista correspondiente a la categoría seleccionada.
    - **Eliminar boleto:** Cuando un boleto es vendido, su estado cambia a "vendido" o se elimina si ya no está disponible.
    - **Buscar boleto:** Para verificar si un asiento está disponible, se recorre la lista para encontrar el boleto correspondiente al asiento seleccionado.

### 2. Matrices de Asientos: 10 pts

Se utilizarán **matrices** para representar los asientos del estadio de manera visual y gestionar su disponibilidad.

- **Objetivo:** Representar los asientos disponibles y ocupados de manera visual en la interfaz gráfica.
- **Cómo se utiliza:**

- Se usará una matriz bidimensional (un arreglo de arreglos) para simular el mapa de asientos. Cada celda de la matriz representará un asiento en el estadio.
- El valor de cada celda será **True** si el asiento está ocupado y **False** si está disponible.
- Al seleccionar un asiento, el sistema verifica la matriz para asegurarse de que esté disponible y luego lo marca como ocupado (True) al realizar la compra.

### 3. HashMap para Precios:

10 pts

Se utilizará un **HashMap** para asociar las categorías de boletos con sus respectivos precios.

- **Objetivo:** Asociar de forma eficiente las categorías de boletos (VIP, General, Preferencial) con sus precios correspondientes.
- **Cómo se utiliza:**
  - La **clave** del mapa será el nombre de la categoría (por ejemplo, "VIP", "General", "Preferencial").
  - El **valor** será el precio asociado a cada categoría.
  - El sistema utilizará este mapa para obtener el precio correspondiente al seleccionar una categoría de boleto.
- **Operaciones:**
  - **Obtener precio:** Cuando el usuario selecciona una categoría, el sistema consulta el mapa para obtener el precio de esa categoría.
  - **Actualizar precio:** Si es necesario, el sistema puede actualizar el precio de una categoría modificando el valor asociado a la clave correspondiente en el mapa.

### 4. Cola (FIFO) para Reportes de Ventas:

10 pts

Se utilizará una **cola** para gestionar los reportes de ventas de boletos, almacenándolos en orden de generación.

- **Objetivo:** Almacenar los reportes de ventas generados en el orden en que se completan las compras.
- **Cómo se utiliza:**
  - Cada venta genera un reporte que incluye la fecha, los boletos vendidos, los ingresos y detalles de los asientos ocupados. Este reporte se almacena en una **cola**.
  - **Operaciones:**
    - **Encolar reporte:** Después de cada compra, se genera un reporte de venta que se agrega al final de la cola.
    - **Desencolar reporte:** Al finalizar el día o al consultar los reportes, los reportes se desencolan en el orden en que se generaron.

## 5. Archivos de Texto para Almacenamiento de Reportes:

**10 pts**

Los reportes de ventas se guardarán en **archivos de texto** en formato .txt.

- **Objetivo:** Guardar de manera persistente los detalles de cada venta para su consulta posterior.
- **Cómo se utiliza:**
  - Después de cada compra, el sistema crea un archivo de texto con el reporte de la venta. El archivo tendrá el formato reporte\_ventas\_ddmmaaaa.txt.
  - El archivo contendrá la información sobre la fecha de la venta, la categoría de boleto, los ingresos generados y los detalles de los boletos vendidos.
  - Los archivos de texto se guardarán en el sistema, permitiendo su consulta y análisis en el futuro.

## Entregables:

### 1. Documento

10pts

- Introducción y descripción del problema a resolver.
  - Esquema del diseño.
  - Capturas de pantalla de ejecución y pruebas realizadas.
  - Código Fuente
2. Exposición del proyecto ante la clase, demostrando su funcionalidad y justificando las decisiones de diseño.

## Flujo de Trabajo del Sistema:

1. **Selección de categoría y asientos:** El usuario selecciona la categoría de boleto (VIP, General o Preferencial). Luego, la interfaz gráfica muestra una matriz de asientos disponibles, donde el usuario elige el asiento deseado.
2. **Confirmación de compra:** El sistema calcula el precio total de la compra, según la categoría y los asientos seleccionados, y luego confirma la compra.
3. **Generación de reportes:** Después de cada compra, se genera un reporte de venta que se agrega a la cola de reportes y se guarda en un archivo de texto.
4. **Almacenamiento de datos:** Todos los boletos vendidos y las categorías de boletos se almacenan en las listas enlazadas. Los reportes de ventas se guardan en archivos de texto.

## Resumen de las Estructuras de Datos:

- **Listas Enlazadas:** Para gestionar los boletos disponibles y vendidos por categoría.
- **Matrices:** Para representar visualmente los asientos del estadio y su disponibilidad.

- **HashMap:** Para asociar las categorías de boletos con sus precios.
- **Cola (FIFO):** Para almacenar los reportes de ventas en el orden en que se generan.
- **Archivos de Texto:** Para guardar los reportes de ventas de manera persistente.

Este diseño permitirá una gestión eficiente de los boletos, una experiencia de compra fácil para los usuarios y la generación de reportes de ventas para el análisis posterior.