

Materia:	Programación II		
Nivel:	2º Cuatrimestre		
Tipo de Examen:	Recuperatorio Primer Parcial		
Apellido ⁽¹⁾ :		Fecha:	17 jun 2025
Nombre/s ⁽¹⁾ :		Docente a cargo ⁽²⁾ :	Baus / Rocha
División ⁽¹⁾ :	122-1	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.

Sistema de Gestión de Espectáculos en Vivo

Una empresa de entretenimiento gestiona distintos tipos de espectáculos en vivo, tales como conciertos, obras teatrales y festivales. Cada espectáculo tiene un **nombre**, una **fecha(LocalDate)** y una **duración** en minutos.

Los **conciertos** incluyen un **género musical** (POP, ROCK, JAZZ) y una lista de **artistas(String)** que participan.

Las **obras teatrales** tienen una **clasificación por edad** (ATP, MAYOR13, MAYOR18) y el **nombre del director**.

Los **festivales** indican la **cantidad de escenarios** disponibles y si cuentan o no con **zona de camping**.

Los **conciertos** y **festivales** pueden ser **transmitidos en vivo**, y deben implementar el método `transmitir()` de la interfaz `Transmitible`.

Los **conciertos** y **obras teatrales** pueden ser **calificados por el público**, implementando el método `calificar(int puntaje)` de la interfaz `Calificable`.

Funcionalidades requeridas

Se requiere desarrollar una clase `GestorEspectaculos` que contenga una lista de espectáculos y permita realizar las siguientes operaciones:

- `agregarEspectaculo(Espectaculo e)`: agrega un espectáculo al sistema. Se debe lanzar una **excepción personalizada** si ya existe otro espectáculo con el mismo nombre y la misma fecha.
- `mostrarEspectaculos()`: muestra todos los espectáculos registrados en el sistema, incluyendo los atributos comunes y los específicos de cada tipo.
- `transmitirEventos()`: inicia la transmisión de todos los espectáculos que pueden hacerlo (conciertos y festivales), e informa cuáles no pueden ser transmitidos.
- `filtrarPorGenero(Genero g)`: devuelve e imprime todos los conciertos que pertenecen al género especificado.
- `calificarEvento(String nombre, int puntaje)`: permite calificar un espectáculo por su nombre. Si no es calificable, se debe informar al usuario.

A partir del enunciado anterior, se solicita:

1. Realizar el diagrama de clases completo en umletino.
2. Implementar el código fuente en Java que resuelva las funcionalidades solicitadas.

Criterios de corrección:

1. Aprobación:

- El diagrama de clases debe reflejar correctamente la jerarquía de herencia.
- Las clases y atributos deben estar correctamente implementados.
- Los métodos solicitados deben funcionar adecuadamente.
- El manejo básico de excepciones debe estar presente, especialmente al agregar plantas duplicadas.

2. Promoción:

- El diagrama debe incluir detalles como visibilidad de atributos y métodos, así como relaciones adicionales (por ejemplo, composición).
- El código debe seguir principios de POO, con buen manejo de encapsulamiento y visibilidad de atributos.
- Se espera el uso de un enum e interfaz.
- El manejo de excepciones debe ser avanzado, con excepciones personalizadas y mensajes claros.
- Se valora el uso de colecciones eficientes.
- Buena modularización del código.