

Teoría

- 1) ¿Cuál es la diferencia entre usar `isinstance` y `type` para identificar objetos en una jerarquía de clases? ¿Cuál de las dos formas garantiza la reutilización de código? Porqué?
- 2) ¿En qué consiste el mecanismo de resolución de conflictos que implementa Python para la herencia múltiple? ¿Qué problema resuelve?

Práctica

Contexto

Un teatro independiente desea informatizar su sistema de programación. Los espectáculos ofrecidos pueden ser de dos tipos:

- Obras de teatro, que incluyen información sobre el nombre del director y la duración (en minutos).
- Conciertos, que incluyen información sobre la banda o artista y la cantidad de músicos participantes.

De todos los espectáculos se registra: nombre, fecha de función y precio base.

El sistema debe permitir calcular el precio final de entrada, que varía según el tipo de espectáculo:

- Para las obras de teatro, el precio final se calcula como:

$$\text{Precio_entrada} = \text{precio_base} + (\text{duracion} * 10)$$

- Para los conciertos, el precio final se calcula como:

$$\text{Precio_entrada} = \text{precio_base} + (\text{cantidad_musicos} * 500)$$

El analista le solicita a usted que desarrolle una aplicación con las siguientes restricciones.

- a) Definir la jerarquía de clases para representar los espectáculos, con sus atributos y métodos necesarios.
- b) Definir una clase colección *Espectáculos*, que use una lista Python para almacenar los objetos creados de la jerarquía planteada.
- c) Implementar un programa principal con un menú de opciones que permita testear las siguientes acciones:
 1. Agregar los espectáculos a la colección. (*Validar que el objeto que se agrega sea un espectáculo*, usar `isinstance`). Lanzar la excepción `ValueError`, si se intenta agregar un espectáculo para una fecha que ya está registrada.
 2. Mostrar para todos los espectáculos el nombre del espectáculo, fecha y el precio final de la entrada. Este método debe ser definido en la superclase. La superclase debe incluir un método abstracto, que luego debe redefinirse en las subclases. Estas implementaciones deben garantizar la reutilización de código (NO DEBEN REPETIRSE SENTENCIAS).
 3. Listar aquellos espectáculos cuyo precio de entrada sea superior a \$10.000.
 4. Leer por teclado una cantidad de músicos y mostrar nombre, fecha y precio final de la entrada del espectáculo cuya cantidad de músicos participantes supera la cantidad dada.