

Contexto - Grupo de Teatro

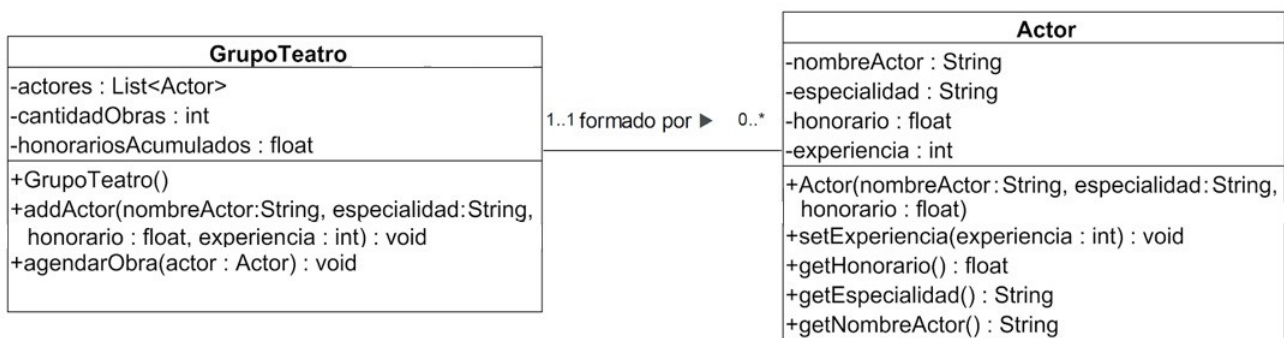
Eleonora es profesora de la Escuela Municipal de Teatro y con los alumnos más avanzados de la carrera del profesorado de teatro quiere crear grupos de teatro y salir de gira por los barrios. Hasta el momento Eleonora utiliza sus boletines de cartón blanco que rellena cada trimestre a mano. Sin embargo, dado el éxito de la escuela, cada vez más son los alumnos que se inscriben no solo de la ciudad, sino de varias ciudades y pueblos de alrededor. Por esta razón, Eleonora necesita desarrollar un sistema que la ayude en el manejo de los alumnos y grupos de teatro.

Como sabe que cursamos el programa 111Mil se puso en contacto con nosotros para que la ayudemos a construir el sistema, que en esta primera etapa del proyecto deberá registrar los actores y obras a realizarse en diferentes lugares.

Ejercicio 1 - Implementar desde el diagrama de clases

Dado que Eleonora es experta en modelado de software, ha creado un diagrama de clases UML para indicarnos lo que se desea. Otros programadores comenzaron con la codificación, pero nos piden que completemos el código para:

- Declarar las variables actores, cantidadObras y honorariosAcumulados de la clase GrupoTeatro.
- Implementar el método addActor de la clase GrupoTeatro. El mismo deberá agregar un nuevo actor a la lista actores.
- Implementar los métodos setExperiencia y getHonorario de la clase Actor.



```
public class GrupoTeatro{
```

//a. Declarar las variables actores, cantidadObras y honorariosAcumulados

```
    public GrupoTeatro() {  
        actores= new ArrayList<Actor>();  
        cantidadObras = 0;  
        honorariosAcumulados = 0;  
    }  
  
    public void agendarObra(Actor actor){  
        for (Actor miActor : actores)  
            if(miActor.getNombreActor().equals(actor.getNombreActor())){  
                cantidadObras++;  
                honorariosAcumulados = honorariosAcumulados + actor.getHonorario();  
            }  
    }  
}
```

//b. Implementar el método addActor

```
}
```

```
public class Actor{  
    private String nombreActor;  
    private String especialidad;  
    private float honorario;  
    private int experiencia;  
  
    public Actor(String nombreActor, String especialidad, float honorario) {  
        this.nombreActor = nombreActor;  
        this.especialidad = especialidad;  
        this.honorario = honorario;  
    }  
    public String getNombreActor() {  
        return this.nombreActor;  
    }  
}
```

//c. Implementar los métodos setExperiencia y getHonorario

```
}
```

- a) Declarar las variables actores, cantidadObras y honorariosAcumulados de la clase GrupoTeatro.
- b) Implementar el método addActor de la clase GrupoTeatro. El mismo deberá agregar un nuevo actor a la lista actores.

c) Implementar los métodos setExperiencia y getHonorario de la clase Actor.

Ejercicio 2 - Implementar un método a partir de un enunciado

Programar en Java el código del método obtenerCantidadEspecialidad en la clase GrupoTeatro. El mismo recibirá una variable llamada especialidad de tipo String y retornará la cantidad de dichos actores que tienen esa especialidad en el grupo. En caso de no encontrar ninguna especialidad deberá retornar cero.

Ejercicio 3 - Interpretación de código

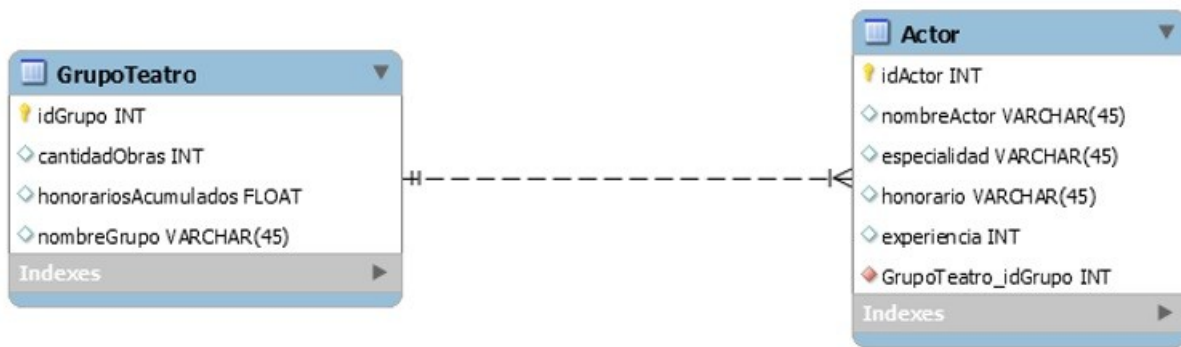
Un desarrollador amigo de Eleonora implementó el siguiente método en la clase GrupoTeatro pero no usó nombres representativos. Indique cuál de las siguientes opciones explica lo que hace el código:

```
public float xxxx() {  
    float honorarios = 0;  
    int cantActores = actores.size();  
    for (int i = 0; i < cantActores ; i++) {  
        honorarios = honorarios + actores.get(i).getHonorario();  
    }  
    if (cantActores > 0)  
        return honorarios / cantActores ;  
    else return 0;  
  
}
```

- A. Calcula el total de actores del grupo de teatro
- B. Calcula el total de honorarios de los actores del grupo
- C. Calcula el promedio de los honorarios de los actores del grupo
- D. Ninguna de las anteriores.

Ejercicio 4 - Interpretación de Diagrama de Entidades y Relaciones (DER).

A partir del DER, responda Verdadero (V) o Falso (F) a las siguientes afirmaciones (asignar a cada afirmación un valor V o F. Todas las afirmaciones tienen que tener un valor asignado. Puede haber varios V y varios F):



VERDADERO / FALSO

- A- El atributo 'experiencia' en la tabla Actor es del tipo VARCHAR
 - B- El atributo 'honorariosAcumulados' pertenece a la tabla Actor
 - C- Entre las tablas GrupoTeatro y Actor existe una relación N a N
 - D- La clave primaria de la tabla Actor es nombreActor
 - E- La clave foránea de la tabla Actor es GrupoTeatro_idGrupo
 - F- La tabla GrupoTeatro no tiene clave primaria
 - G- Entre las tablas GrupoTeatro y Actor existe una relación 1 a N
- Repetición de las opciones A-G:
- A- El atributo 'experiencia' en la tabla Actor es del tipo VARCHAR
 - B- El atributo 'honorariosAcumulados' pertenece a la tabla Actor
 - C- Entre las tablas GrupoTeatro y Actor existe una relación N a N
 - D- La clave primaria de la tabla Actor es nombreActor
 - E- La clave foránea de la tabla Actor es GrupoTeatro_idGrupo
 - F- La tabla GrupoTeatro no tiene clave primaria
 - G- Entre las tablas GrupoTeatro y Actor existe una relación 1 a N

Ejercicio 5 - Consultas SQL.

Dado el diagrama de entidad-relación presentado en el ejercicio anterior, escriba una consulta SQL que liste los nombres de los actores que tengan la especialidad "clown" ó "mimo" del grupo "Sala 111Mil". El listado debe estar ordenado en forma ascendente por los nombres de los actores