

# CFGs DESARROLLO DE APLICACIONES MULTIPLATAFORMA / WEB

## MODELO EXAMEN



### M03B. PROGRAMACIÓN B

NOMBRE: \_\_\_\_\_

DNI y FIRMA: \_\_\_\_\_

*El presente examen es un modelo de años anteriores. Las preguntas y el formato pueden variar.*



## UF4: Programación orientada a objetos. Fundamentos.

1. Indica 5 palabras reservadas, que no sean modificadores de Java, y descríbelas.

2. Rellena la tabla con SI o NO, dependiendo de la visibilidad que indica el modificador.

MODIFICADOR	CLASE	SUBCLASE	PACKAGE	OTRO
PUBLIC				
PROTECTED				
PRIVATE				
NO ESPECIFICADO				

3. Crea una clase llamada libro. Incluye los atributos que consideres, mínimo 5, para esta clase. También debes incluir los constructores, métodos getter y setter y la sobrecarga del método toString() que devuelva una cadena con la información necesaria. Ejemplo: [Título – Autor]





## UF5: Programación orientada a objetos. Librerías de clases fundamentales.

1. ¿Qué es un formulario en el entorno gráfico?
2. ¿Qué es AWT?
3. ¿Qué es el polimorfismo?
4. Cree el paquete de nombre película y realice el siguiente código:
  - a. La interfaz IPelícula, teniendo en cuenta que el tipo es comparable. Los métodos se explican en el tercer ejercicio.
  - b. El tipo enumerado Género con los valores COMEDIA, DRAMA, TERROR, CIENCIA FICCIÓN, AVENTURAS, INDEFINIDO.
- Cree la interfaz IPeliculaOnline. Los métodos se explican en el tercer ejercicio.
- Crea las siguientes clases basándote en la descripción que se detalla a continuación:

***Película***

**1.1. Propiedades:**

- Nombre, de tipo String, sólo consultable.
- Nacionalidad, de tipo String, sólo consultable.
- Género, de tipo Género, sólo consultable. Los valores que puede tomar son: COMEDIA, DRAMA, TERROR, CIENCIA FICCIÓN, AVENTURAS, INDEFINIDO.
- Presupuesto, de tipo Double, consultable y modificable.
- Año, de tipo Integer, sólo consultable.

**1.2. Constructores:**

- Constructor por defecto sin argumentos.
- Constructor que recibe un parámetro por cada propiedad.

**1.3. Getter's y setter's.** Implemente los métodos necesarios para consultar y modificar los parámetros.**1.4. Criterio de igualdad:** Dos películas son iguales si tienen el mismo nombre y el mismo año. (Utilizar los tipos Comparable y Comparator)**1.5. Orden natural:** Por orden, y si coinciden, entonces por año.**1.6. Representación como cadena:** el nombre, seguido del año entre paréntesis. Ejemplo: Casablanca (1942)**1.7. Restricciones:**

- Solo se permiten fechas posteriores a 1894. Si se incumple esta restricción se lanzará la excepción `IllegalArgumentException`.
- Solo se permiten presupuestos positivos o cero. Si se incumple esta restricción se indicará al usuario que no es posible ese valor para el atributo presupuesto.

***PeliculaOnline*** Es un subtipo de película.

**1.1. Propiedades adicionales:**

- Link en YouTube, de tipo String, consultable y modificable.
- Puntuaciones, de tipo `List<Integer>`, sólo consultable.
- Puntuación media, de tipo Double, propiedad derivada, es decir, es calculada.

**1.2. Constructores:**

- Un único constructor, con un parámetro por cada propiedad básica del tipo, exceptuando la propiedad puntuaciones.

**1.3. Getter's y setter's.** Implemente los métodos necesarios para consultar y modificar los parámetros.

- 1.4. Criterio de igualdad: el mismo que el de Película.
- 1.5. Orden natural: el mismo que el de Película.
- 1.6. Representación como cadena: la misma que la de Película.
- 1.7. Añadir puntuación, devuelve void. Añade una nueva puntuación de un usuario (entre 0 y 10).
- 1.8. Restricciones:
  - Si la puntuación no está entre 0 y 10 o es nula, se lanzará la excepción `IllegalArgumentException`.









## UF6: Programación orientada a objetos. Introducción a la persistencia en BD.

1. ¿Qué sentencia necesitamos para conectarnos a una base de datos?  
¿Con qué método podemos ejecutar una sentencia SQL?
2. ¿Cómo podemos conectar nuestra aplicación java a una base de datos?
3. ¿Cómo podemos insertar datos en la base de datos?
4. ¿Cómo podemos recuperar todos los usuarios de nuestra base de datos desde una aplicación java?