

**EVALUACIÓN 2**  
**PROGRAMACIÓN ORIENTADA A OBJETOS - SECCIÓN 1**  
Docente Jazna Meza Hidalgo  
Noviembre 2020

## FORMATO Y CONDICIONES DE ENTREGA

- **Plazo de entrega** Viernes 13 antes de las 23:00
- **Dónde se entrega** En el enlace disponible en la plataforma. Una entrega por grupo, indicando en el mensaje CLARAMENTE el nombre del compañera(o) y la sección a la que pertenece.
- **LAS COPIAS SON EVALUADAS CON LA NOTA MÍNIMA**

## DIAGRAMA DE CLASES

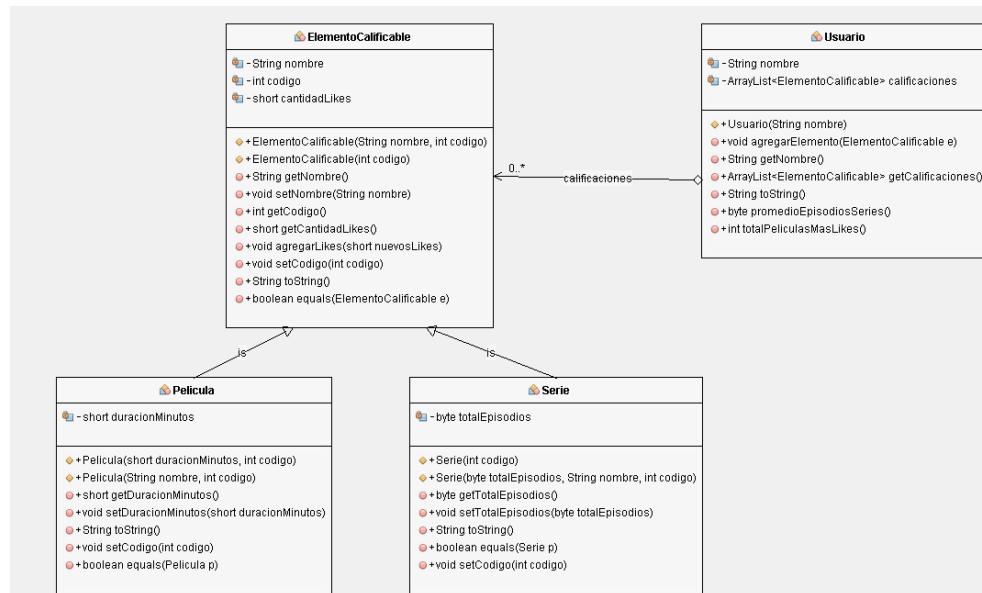


Figure 1: Diagrama de clases

## REGLAS DE NEGOCIO

- El código de las series debe terminar con 300.
- El código de las películas debe comenzar con 200.

- La cantidad de likes debe ser un valor mayor o igual a cero.

## CRITERIOS DE IGUALDAD DE OBJETOS

- Dos series son iguales si su código es el mismo y su cantidad de episodios es la misma.
- Dos películas son iguales si su código es el mismo y su duración en minutos es la misma.

## PROBANDO LAS CLASES

Construir una clase llamada Main que permita cumplir con los siguientes requerimientos:

- Crear 1 objeto de la clase Usuario.
- Crear 2 objetos de la clase Serie.
- Crear 2 objetos de la clase Película.
- Agregar los 4 elementos de calificación al usuario (usando el método creado para ello).
- Agregar likes a una de las series y a 2 películas.
- Imprimir, usando el método toString() todos los objetos creados (ver formato en la sección formato de salida)
- Imprimir, usando el método especializado, el promedio de la cantidad de episodios de las series asociadas al usuario.
- Imprimir, usando el método especializado, el total de películas con mayor cantidad de likes asociadas al usuario.

## FORMATO DE SALIDA

```
{Jazna} tiene 4 series/peliculas calificadas  
Avengers. End game tiene [850] likes con [120] minutos de duración  
Salvation Temporada 1 tiene [100] likes con [16] episodios
```

Figure 2: Formato de impresión de los objetos