

Actividades UD3 (Festival)

Clase Persona

Representa a las personas que asisten al festival.

Atributos (privados):

- nombre → *String*, mínimo 3 caracteres.
Si tiene menos de 3, se añadirá "...".
- edad → *Number*, mínimo 18 años.
Si se introduce un valor inferior, se ajustará automáticamente a 18.
- genero → Debe ser uno de los siguientes valores:
"hombre", "mujer", "otro".

Métodos:

- Constructor (nombre, edad, genero)
- *toString()* → Devuelve toda la información.
- Getters y Setters con validaciones.

Clase Jurado (hereda de Persona)

Representa a los miembros del jurado del festival.

Atributos adicionales (privados):

- profesion → *String*, mínimo 5 caracteres.

Métodos:

- Constructor (nombre, edad, genero, profesion)
- *toString()* → Incluye toda la información heredada y propia.
- Getters y Setters con validaciones.

Clase Público (hereda de Persona)

Representa al público general que asiste al festival.

Atributos adicionales (privados):

- entrada → *String*, con posibles valores "general", "vip", "estudiante".
Si se introduce otro valor, se lanzará un error.
- puntuacion → *Number*, entre 0 y 10.
Si no está en ese rango, se lanzará un error.

Métodos:

- Constructor (nombre, edad, genero, entrada, puntuacion)
- *toString()* → Incluye toda la información heredada y propia.
- Getters y Setters con validaciones.

Actividades UD3 (Festival)

Clase Produccion

Representa las películas o cortometrajes que participan en el festival.

Atributos (privados):

- titulo → *String*.
- nacionalidad → *String*.
- genero → *String*, con la primera letra en mayúscula y el resto en minúsculas (se corrige automáticamente).
- anio → *Number*, no puede ser inferior a 1999.

Métodos:

- Constructor (titulo, nacionalidad, genero, anio)
- *toString()* → Devuelve toda la información.
- Getters y Setters con validaciones.

Clase Festival

Contiene toda la información del festival.

Atributos (privados):

- nombre → *String*, no en blanco y al menos 3 caracteres.
- ciudad → *String*, no en blanco.
- edicion → *Number*, no negativa.
- descripcion → *String*, máximo 300 caracteres.
- aProducciones → *Array* que almacena objetos de tipo Produccion (sin duplicados).
- aPersonas → *Array bidimensional*:
 - Fila 0 → Público.
 - Fila 1 → Jurado.

Métodos:

- Constructor (nombre, ciudad, edicion, descripcion)
- introducirProduccion(ObjetoProduccion)
 - Añade la producción si no existía antes (mismo título y año). Devuelve true si se añade, false si estaba repetida.
- eliminarProduccion(titulo)
 - Elimina las producciones con ese título. Devuelve true o false.
- anadirPersona(ObjetoPersona)
 - Si es Jurado, se añade a la fila 1.
 - Si es Publico, se añade a la fila 0.
 - En caso contrario, lanza error.
- *toString()* → Devuelve la información general del festival.
- Getters y Setters correspondientes.

Creación y uso de objetos

Crea un script principal donde:

1. Crees al menos:
 - 3 objetos de tipo Público.
 - 2 objetos de tipo Jurado.
 - 2 producciones.
 - 1 festival.
 2. Añadas los objetos creados al festival mediante sus métodos (`anadirPersona`, `introducirProduccion`).
 3. Muestres por consola:
 - La información del festival.
 - La lista de producciones.
 - Los datos del jurado.
 - Los datos del público.
-

Consideraciones generales

- Cada clase debe ir en un **módulo independiente (archivo .js)**.
- Se debe usar la **sintaxis ES6** (class, import/export).
- Todos los atributos deben ser **privados** (#atributo).
- Las validaciones deben realizarse en los **setters**.
 - Si un valor no cumple las condiciones, debe **lanzarse una excepción (throw)**.
- La creación de objetos debe realizarse dentro de **bloques try...catch**, mostrando en consola los posibles errores.
- Se deben usar las **funciones nativas de arrays** (forEach, some, filter, etc.).
- Se recomienda el uso de **funciones flecha** para simplificar callbacks.
- En el archivo principal (main.js o index.js) se probarán todos los métodos.