

Cuentos infantiles.

Ejercicio 1: Implementa la clase Personaje para representar a los distintos personajes del cuento.

- **Constructor (nombre, tipo, edad)**
- **nombre:** String. Máximo 10 caracteres. Si tiene más, se añade "...".
- **tipo:** String. Ej: "Hada", "Duende", "Caballero", etc. No debe estar vacío.
- **edad:** Number. No puede ser menor de 5 años. Si es menor, se le asigna 5.
- **toString()** Devuelve toda la información del constructor.
- Sus **setters y getters** correspondientes.

Ejercicio 2: Implementa la clase Protagonista para representar al personaje principal del cuento. Hereda de **Personaje** sus atributos y añade:

- **Constructor (nombre, tipo, edad, cualidad)**
- **cualidad:** String. Describe una habilidad o cualidad única del protagonista, con un mínimo de 5 caracteres.
- **toString()** Devuelve toda la información del constructor.
- Sus **setters y getters** correspondientes.

Ejercicio 3: Implementa la clase Escenario para representar el lugar donde ocurren los eventos.

- **Constructor (nombre, ubicacion, clima)**
- **nombre:** String. Nombre del lugar (Ej: "Bosque Mágico"). Mínimo 3 caracteres.
- **ubicacion:** String. La ubicación del escenario (Ej: "En lo alto de la colina"). No debe estar vacío.
- **clima:** String. Describe el clima, almacenado con la primera letra en mayúscula y el resto en minúsculas.
- **toString()** Devuelve toda la información del constructor.
- Sus **setters y getters** correspondientes.

Ejercicio 4: Implementa la clase Cuento para almacenar los datos de un cuento infantil.

Constructor (titulo, autor, moraleja, descripcion)

- **titulo:** El nombre del cuento, que debe ser un string no inferior a tres caracteres.
- **autor:** String, nombre del autor del cuento, no puede estar en blanco.
- **moraleja:** String. Moraleja del cuento, con un máximo de 100 caracteres.
- **descripcion:** String. Descripción breve del cuento, no superior a 300 caracteres.
- **aEscenarios:** Será un array donde se almacenarán los escenarios. Solo puede contener objetos de tipo **Escenario**.
- **introducirEscenario(objetoEscenario):** Añade el escenario si no estaba anteriormente. No permite repetidos. Devuelve true si se ha insertado y false si no.
- **eliminarEscenario(nombre):** Busca por nombre y elimina el escenario que lo tenga. Devuelve true si se ha eliminado y false si no lo ha podido eliminar.
- **aPersonajes:** Array donde se almacenan personajes y protagonistas.
- **anadirPersonaje():** Añade un personaje o un protagonista. Si es protagonista, se agrega en la fila 1 del array; si es personaje, en la fila 0.
- **toString()** Devuelve la información del título, autor, moraleja y descripción.
- Sus **setters y getters** correspondientes.

Ejercicio 5: Creación y uso de objetos.

- Crea al menos 3 personajes, 3 protagonistas, 2 escenarios y un cuento que use los métodos de añadir personajes, protagonistas y escenarios.
- Muestra la información completa del cuento de la manera que te resulte más cómoda.

Consideraciones generales a tener en cuenta:

- Para organizar el código de la aplicación se utilizarán módulos. Cada clase irá en un módulo independiente.
- Se debe usar la sintaxis ES6 para implementar las clases y objetos.

- Todos los atributos se consideran privados. Crea los métodos get y set para acceder y fijar un valor para los atributos cuando se necesite.
- La comprobación de restricciones sobre los atributos debe realizarse en el método set del atributo. Si no se cumplen las restricciones, se mostrará un error. Para ello lanza una excepción que interrumpa la ejecución del código. Realiza la creación de objetos en un bloque try-catch para informar a través de consola del problema.
- Se deben usar las funciones nativas del lenguaje en referencia a arrays y otros objetos siempre que sea posible.
- Usa la notación flecha para simplificar la sintaxis de las funciones anónimas.