

Fundamentos de Programación

| Proyecto Final IMDB

Índice

1. Creación de la Clase Professional en Typescript
2. Creación de la Clase Movie en Typescript
3. Creación de la Clase Imdb en Typescript
4. Avanzada

| Clase Profesional

Paso 1: Clase Professional

1. El objetivo de este proyecto final es crear una BBDD de películas y guardar todo ello en un fichero en formato .json.
2. Crear un proyecto en GIT llamado **IMDB** para que contenga todos los ficheros del proyecto y compartirlo con todos los desarrolladores del equipo.
3. Crear una clase denominada **Professional** con los siguientes atributos públicos:
 - **name**: string
 - **age**: number
 - **weight**: number
 - **height**: number
 - **isRetired**: boolean
 - **nationality**: string
 - **oscarNumber**: number
 - **profession**: string
4. Esta clase debe tener un **constructor** que incluirá en sus parámetros **TODOS** estos **atributos**
5. Esta clase tendrá un **MÉTODO PUBLICO** que muestre en consola el valor de **TODOS** sus **atributos**

Paso 2: Fichero testProfessional.ts

1. Crear un archivo **testProfessional.ts** con varios objetos de la clase **Professional** y mostrar todos sus atributos usando el **método anterior**.
2. Actualizar el **GIT** con esta nueva versión de la app
3. No olvidad que los ficheros **.js** deben estar en el .gitignore.

| Clase Movie

Paso 3: Clase Movie

1. Crear una clase que se llame **Movie** con los siguientes atributos **Públicos**.
 - title: **string**
 - releaseYear: **number**
 - actors: **Professional[]**
 - nationality: **string**
 - director: **Professional**
 - writer: **Professional**
 - language: **string**
 - plataforma: **string**
 - isMCU: **boolean**
 - mainCharacterName: **string**
 - producer: **string**
 - distributor: **string**
 - genre: **string**
2. Los únicos parámetros del constructor son: **title**, **releaseYear**, **nationality** y **genre**
3. Crear un **método PÚBLICO** que muestre todos los datos de las **películas**

Paso 4: Fichero testMovie.ts

1. Tomando estas clases crear el fichero **testMovie.ts** que muestre todos los datos de las películas.
2. Actualizar el **GIT** con esta nueva versión de la app

| Clase Imdb

Paso 5: Clase Imdb

1. Crear una clase denominada **Imdb** cuyo **constructor** reciba un **array** de **objetos** de la clase **Movie**
2. Debe tener un único **atributo publico** denominado películas que es un array de objetos de la clase **Movie**
3. Crear un fichero **testImdb** para probar esta clase creando un objeto de tipo **Imdb**.
4. Actualizar **GIT** con una nueva versión de la app.

| **Avanzado**

Paso 6: Lectura/Escritura en Fichero

1. Convertir a un string el objeto **Imdb**.
 - **PISTA**: `JSON.stringify()`
2. Guardar dicho objeto en un fichero **JSON** con el nombre "**imdbBBDD.json**".
 - **PISTA**: `fs.writeFileSync(file, data[, options])`
3. Actualizar el GIT con una nueva versión de la app
4. Leer el fichero "**imdbBBDD.json**" y almacenarlo en una instancia de la clase **Imdb**
 - **PISTA**: `fs.readFileSync(path[, options])`
 - **PISTA**: `JSON.parse()`

Paso 7: Añadir nuevos métodos a la clase

Crear los siguientes métodos público en la clase `Imdb`:

- `escribirEnFicheroJSON(nombreFichero: string)`

Toma el nombre de un fichero como parámetro de entrada y genera un fichero `.json` con toda la información guardada en el objeto `Imdb`.

- `obtenerInstanciaIMDB(nombreFichero:string):Imdb`

Devuelve un objeto de la clase `Imdb` creado a partir del fichero.

Paso 8: Leer datos desde la consola

- Crear un nuevo fichero **finaltest.ts** que solicite por pantalla cada uno de las propiedades de la clase **Profesional** e inserte un nuevo profesional a una película del fichero **"imdbBBDD.json"**
 - **PISTA:** Buscar **readLineSync**

codenotch