# **PROYECTO: BBDD HobbyGame**

El objetivo de este proyecto final es crear una BBDD de videojuegos y consolas y guardar todo ello en ficheros.

* Crear un proyecto en GIT llamado HobbyGame para que contenga todos los ficheros del proyecto y compartirlo con todos los desarrolladores del equipo.
* Crear una clase denominada Person con los siguientes atributos privados:
  + name
  + nationality
  + profession. Puede ser developer, tester, manager o director
  + experienceYears. Años de experiencia
* Esta clase debe tener un constructor que incluirá en sus parámetros todos estos atributos.
* Metodos setters y getters de los atributos privados.
* Crear un método que te devuelva verdadero si la persona pasada como parámetro es igual a la persona.
* Esta clase tendrá un método que muestre en consola el valor de todos estos atributos
* Esta clase tendrá un método que devuelva en un string el valor de todos estos atributos.
* Crear un archivo testPerson y con varios objetos de la clase Person y mostrar todos sus atributos usando los métodos anteriores.
* Actualizar el GIT con una nueva versión de la app
* Crear una clase denominada Platform con los siguientes atributos privados:
  + name
  + releaseYear
  + company.
  + nBits.
  + generation.
  + hasHD
  + hdSize
  + hasInternetConnetion
  + storageDevice. Puede ser (cartucho, tarjeta, cd, dvd o bluray)
  + color.
* Esta clase debe tener un constructor que incluirá en sus parámetros todos estos atributos.
* Métodos getters y setters
* Crear un método que te devuelva verdadero si la plataforma pasada como parámetro es igual a la plataforma.
* Esta clase tendrá un método que muestre en consola el valor de todos estos atributos
* Esta clase tendrá un método que devuelva en un string el valor de todos estos atributos.
* Crear un archivo testPlatform y con varios objetos de la clase Platform y mostrar todos sus atributos usando los métodos anteriores.
* Actualizar el GIT con una nueva versión de la app
* Crear una clase que se llame Videogame con los siguientes atributos privados
  + title
  + releaseYear
  + developers: array of Person
  + nacionality
  + director: Person
  + languages: array of string
  + plataforms: array of platform
  + price
  + score: (0-10)
* Esta clase debe tener un constructor que incluirá en sus parámetros todos estos atributos.
* Métodos getters y setters
* Crear un método que te devuelva si el videojuego ha salido para una platafoma determinada.
* Crear un método que te devuelva si un desarrollador ha desarrollado el videojuego.
* Crear un método que te devuelva si el videojuego ha sido desarrollado en un idioma determinado.
* Crear un método que muestre todos los datos de las películas
* Crear un método que devuelva en un string el valor de todos estos atributos.
* Tomando estas clases crear un fichero testVideoGame que muestre todos los datos de las películas y pruebe todos sus métodos.
* Actualizar GIT con una nueva versión de la app
* Crear una clase denominada HobbyGame cuyo constructor reciba un array de objetos de la clase Videogame
* Con un único atributo privado que es un array de objetos de la clase Videogame
* Crear los métodos getter y setters necesarios.
* Crear un metodo que devuelva el precio total de todos los videojuegos de una plataforma determinada.
* Crear un método que devuelva el precio total de los videojuegos de la biblioteca.
* Crear un método que devuelva los videojuegos desarrollados en un idioma determinado.
* Crear un método que devuelva los videojuegos desarrollados por un desarrollador
* Crear un método que te devuelva los videojuegos desarrollados para una determinada plataforma y en un determinado idioma.
* Crear un método que devuelva todos los videojuegos cuya nota sea mayor o igual a 5.
* Crear un fichero testHobbyGame para probar esta clase
* Actualizar GIT con una nueva versión de la app
* Convertir a JSON el objeto hobbyGame
* Guardar dicho objeto en un fichero JSON con el nombre “hobbyGameBBD.json”.
* Actualizar el GIT con una nueva versión de la app
* Leer el fichero “hobbyGameBBDD.json” y almacenarlo en una instancia de la clase HobbyGame
* Crear un método en la clase HobbyGame denominado *writeOnJSONFile(fileName: string)* que tome el nombre de un fichero como parámetro de entrada y genere un fichero json con toda la información de la base de datos. Hacer todas las modificaciones necesarias en las clases necesarias.
* Crear un método de la clase HobbyGame denominado *getInstance(nombreFichero:string):HobbyGame* que devuelva un objeto de la clase HobbyGame creado a partir del fichero.
* Crear un nuevo fichero testHobbyGameJson que pruebe estos métodos.
* Crear un nuevo fichero testInteractive que solicite por pantalla cada uno de las propiedades de la clase Videogame e inserte una nueva película en el fichero “hobbyGameBBDD.json”