

Trabajo Practico: Héroes y villanos

Julian Gasparini, Valdez Gustavo

Algoritmos y programación II – 1er cuatrimestre 2022

Universidad Nacional de Tres De Febrero.

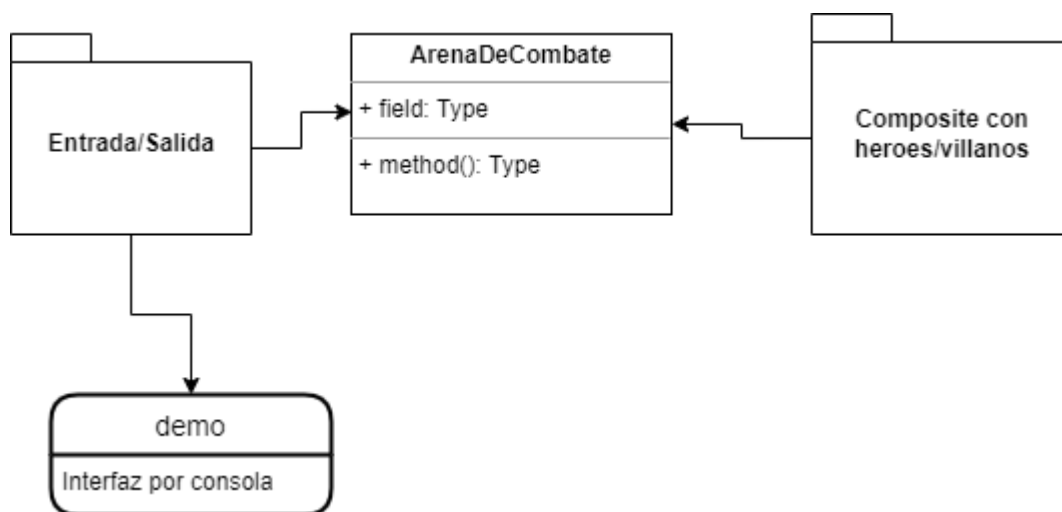
[Repositorio](#) al TP.

Planificación Inicial:

Al comenzar a trabajar en el TP dos grandes problemas:

- Modelar un mecanismo en donde dos clases distintas, pero con un funcionamiento similar.
- Trabajar con distintas formas de entrada de datos para almacenar y controlar el comportamiento de dichas clases.

Nuestro primer planteo fue el siguiente:



ManejoDePersonajes:

Para el manejo de personajes se optó por implementar un patrón de diseño composite. Las clases Personaje y Liga ambos heredan los métodos y atributos de Competidor. La liga a su vez, tiene un método para agregar Personajes y otra Ligas a una lista de Competidores.

Algunas de nuestras dificultades fueron:

- Que vuelva al inicio al comparar con la última posición del enum de Características.
- Para ordenar los competidores se tuvo que hacer una modificación del algoritmo de QuickSort para que ordene correctamente por la característica que elijamos.

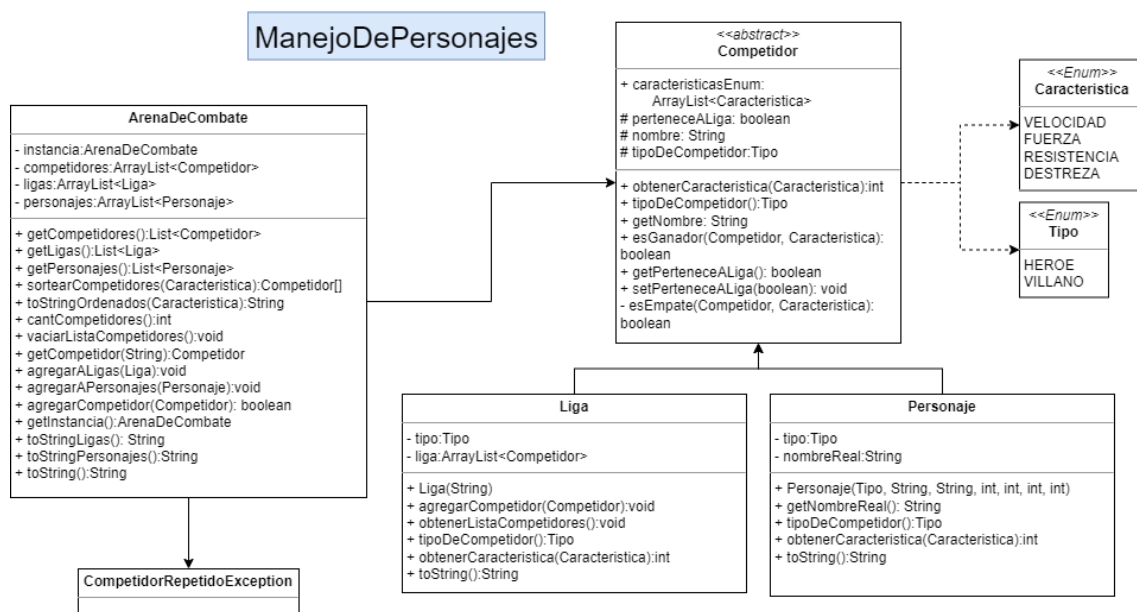


Ilustración 1- UML del package ManejoDePersonajes

Package IO:

Para la entrada y salida de datos se diseñó un package donde dos clases “CreadorPersonajes” y “CreadorLiga” llevan la responsabilidad de instanciar correctamente los objetos en memoria que luego el resto del programa utilizará para desarrollar sus funciones. También cuenta con una clase **Escritor** que es la encargada de escribir los datos almacenados en archivos de texto para que puedan ser almacenados en la memoria estática. Y, por último, una clase **Menu** que se utiliza para facilitar el código de nuestra clase **main**, que es desde donde se ejecuta nuestro programa. Al desarrollar este package encontramos muchos problemas en el manejo de excepciones, las cuales muchas finalmente terminaron de controlarse en el método

main de la clase Demo. Consideramos que este aspecto del programa puede mejorarse, pero debido a la falta de tiempo y de personas no pudimos lograr el resultado que queríamos. También nos encontramos modificando la clase CreadorPersonajes y CreadorLigas, la cual inicialmente estaba planteada como una única clase Lector, pero decidimos separarla debido a que la clase abarcaba mucha responsabilidad dentro del programa y esto no es una buena aplicación de POO.

IO

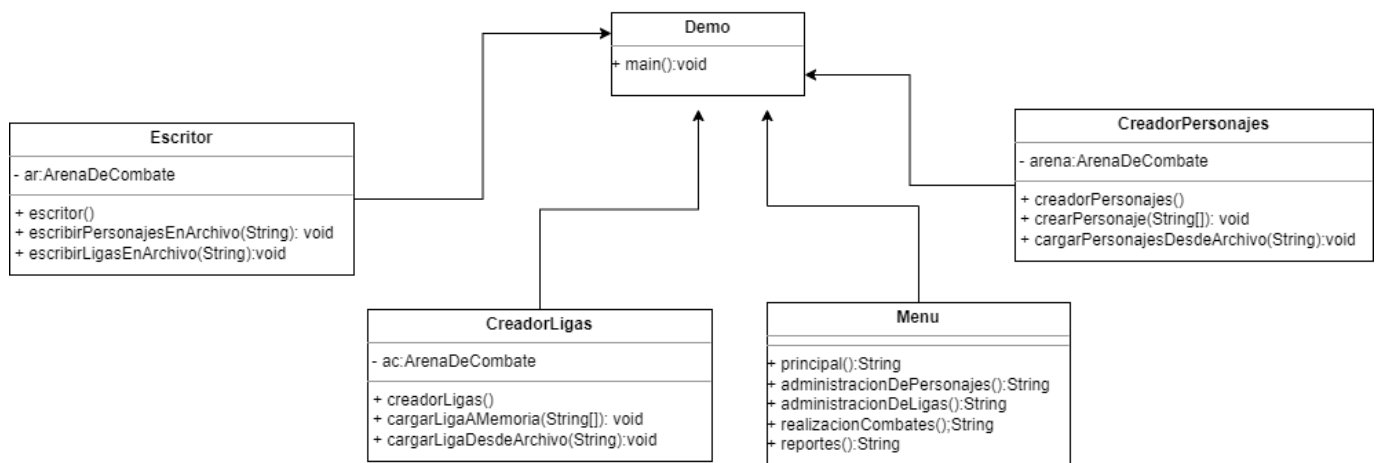


Ilustración 2 - UML del package IO

Como conclusión, reconocemos que nuestro trabajo esta lejos de ser perfecto incluso para nuestro nivel de conocimiento. Sin embargo, afirmamos que es capaz de resolver los principales problemas planteados en la consigna del TP.