

# HÉROES Y VILLANOS



2023  
Paradigmas de programación

Estevez, Martín  
Pucci, Federico  
Segura, Franco

# Introducción

En este trabajo práctico, se implementó una aplicación en Java que implementa los fundamentos de la Programación Orientada a Objetos (POO), el uso de estructuras de datos abstractas (TDA), el uso de archivos, también se implementan patrones de diseño y un menú por consola.

La aplicación tiene como objetivo realizar batallas entre héroes, villanos y ligas de héroes o villanos. El usuario puede crear personajes, ligas y batallas. También puede consultar la información de cada uno de estos elementos, así como también reportes finales

## Objetivos específicos

Los objetivos específicos de este trabajo práctico se cumplieron satisfactoriamente. Se implementaron clases para representar los diferentes elementos de la aplicación, se implementaron estructuras de datos abstractas para representar las colecciones de elementos de la aplicación, se implementó un menú por consola que permitió al usuario interactuar con la aplicación y se implementaron las reglas de las batallas entre héroes, villanos y ligas de héroes o villanos mediante las características.

## Requisitos

Los requisitos de este trabajo práctico también se cumplieron satisfactoriamente. La aplicación permite al usuario crear héroes, villanos, ligas y batallas según las características de los personajes. La aplicación implementa las reglas de las batallas entre personajes y ligas de héroes o villanos mediante las reglas descriptas en la consigna.

## Estructura de la aplicación

La aplicación se estructura de la siguiente manera:

- Clases: Las clases representan los diferentes elementos de la aplicación.
- TDA: Las estructuras de datos abstractas representan las colecciones de elementos de la aplicación.
- Menú por consola: El menú por consola permite al usuario interactuar con la aplicación.
- Reglas de las batallas: Las reglas de las batallas determinan cómo se llevan a cabo las batallas entre héroes, villanos y ligas por medio de las características.

## Implementación

La implementación de la aplicación se llevó a cabo en las siguientes etapas:

- Etapa 1: Se diseñaron las clases que representarían los diferentes elementos de la aplicación.
- Etapa 2: Se implementaron las clases diseñadas en la etapa 1.
- Etapa 3: Se implementaron las clases, métodos y estructuras de datos.
- Etapa 4: Se implementó el menú por consola que permitiría al usuario interactuar con la aplicación.
- Etapa 5: Se probó la aplicación.
- Pruebas

La aplicación se probó utilizando los siguientes métodos:

- Pruebas unitarias: Las pruebas unitarias se utilizaron para probar cada una de las clases de la aplicación mediante JUnit.
- Pruebas de integración: Las pruebas de integración se utilizaron para probar la interacción entre las diferentes clases de la aplicación.
- Pruebas de sistema: Las pruebas de sistema se utilizaron para probar la aplicación como un todo.

## Decisiones de diseño

Para la construcción de los personajes y ligas se usan los conceptos típicos vistos en la materia, es interesante el uso de la composición para las características y como se manejan de forma independiente.

Por otro lado, para las ligas se decide tratarlas como si fueran un “personaje” en sí mismo al promediar las características. Esto nos facilitó la implementación de las batallas.

En las batallas, por lo dicho anteriormente, la facilidad que tuvimos fue que, al tratar la liga de esa forma, es que no importa si es un personaje versus otro personaje o liga, o una liga contra otra liga, se manejan de la misma forma.

Vimos que en Java hay varias formas para el manejo de archivos, en nuestro caso, como indica la consigna, usamos la clase Scanner. Para el menú, usamos un switch-case, y, aunque es una interfaz por medio de la consola, a nuestro parecer es bastante simple de utilizar.

## Descripción de archivos

- Archivos.java: En este archivo realizamos la carga y el guardado tanto de personajes como de ligas.
- Batalla.java: En este archivo realizamos la implementación de los tres tipos de batallas, personaje versus otro personaje, personaje versus una liga y una liga versus otra liga.
- Liga.java: En este archivo, realizamos todo el manejo de las ligas, desde la creación, como la modificación hasta el cálculo del promedio de las características para poder realizar batallas.
- Menu.java: En este archivo se maneja toda la parte de la interfaz e interacción con el usuario. Cabe destacar que se agregaron colores a las opciones para hacer el menú más atractivo.
- Personaje.java: En este archivo hicimos todas las funcionalidades básicas para la creación de personajes, también agregamos la búsqueda dentro de los personajes ya cargados en memoria.
- Características.java: Este archivo es bastante simple, solamente se crea la clase características, la cual es usada en los personajes y las ligas.
- ConsoleColors.java: Este es un archivo utilitario donde definimos los colores usados en el menú e interfaz de usuario.
- Reportes.java: En este archivo se implementan las dos características para los reportes indicados en la consigna.
- ComparadorDePersonajePorCaracteristica.java: En este archivo se sobrescribe el método compare para facilitar el manejo de las listas en los reportes.

# Organización y distribución del trabajo

Para organizarnos y distribuirnos el trabajo, debido a la falta de tiempo y poca coincidencia horaria, creamos un repositorio en GitHub en el cual cada uno de los integrantes realizaba las actualizaciones en el mismo según las tareas que realizaba y nos comunicábamos mayormente por Discord, fue un trabajo realizado de forma semi sincrónica, en los momentos en los que coincidíamos con los horarios, organizamos algunas reuniones para comunicar los avances y redistribuir el trabajo restante.

- Federico: Realizó el diseño inicial de las clases y las primeras versiones de estas en la etapa 1. Una vez que eso estuvo listo, colaboró con las etapas 2 y 3. Luego realizó el informe final.
- Franco: Realizó una parte del código de las etapas 2 y 3, y se encargó mayoritariamente de la interfaz de usuario en la etapa 4.
- Martín: Realizó el diagrama UML en la etapa 1. Cuando se finalizó la etapa 1, colaboró con las etapas 2 y 3. Se encargó mayoritariamente del reporting y de las pruebas del sistema.
- Vale destacar que los tres hicimos el mismo aporte para las etapas 2 y 3, pero cada uno se fue destacando en las distintas etapas, ya que en la división del trabajo, elegíamos la tarea que más sencilla nos resultara a cada uno.

## Conclusión

El trabajo es interesante para abarcar todos los conceptos vistos en la materia, ya que se trabajan conceptos básicos como encapsulamiento, herencia y polimorfismo, así como con conceptos más avanzados como son las excepciones y los patrones de diseño. A pesar de que la consigna fue dictada con bastante antelación a la fecha de entrega, debido a la poca disponibilidad horaria de los tres, se nos complicó un poco la organización.