UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 2

CATEDRÁTICO: ING.  DAVID ESTUARDO MORALES AJCOT

TUTOR ACADÉMICO: HERBERTH ABISAI AVILA RUIZ

# MANUAL TÉCNICO

Josué Daniel Fuentes Díaz

CARNÉ: 202300668

SECCIÓN: B+

GUATEMALA, 3 DE MARZO DEL 2,025

# ÍNDICE

## Contenido

# INTRODUCCIÓN

Este manual técnico describe de forma directa la arquitectura, diseño e implementación del software de simulación de torneos de peleas desarrollado en Java. Dirigido a desarrolladores y personal técnico, el documento facilita la comprensión y el mantenimiento del sistema.

El software se compone de tres módulos principales:

- **Carga y Procesamiento:** Se encarga de leer y validar los datos de personajes desde archivos (.lfp) usando la clase LectorArchivo.
- **Simulación del Torneo:** Gestiona las batallas uno contra uno en rondas eliminatorias mediante la clase Torneo.
- **Generación de Reportes:** Crea archivos HTML con las métricas de desempeño (Top 5 de ataque y defensa) a través de la clase ReporteHTML.

Además, el sistema utiliza una interfaz de consola que permite al usuario cargar datos, iniciar el torneo y generar reportes. Este manual detalla las decisiones técnicas y el flujo de interacción entre módulos, ofreciendo una guía clara para futuras mejoras y mantenimiento del código.

# OBJETIVOS

## 1. GENERAL

Elaborar un manual técnico conciso que describa la arquitectura, el diseño e implementación del software de simulación de torneos de peleas en Java, facilitando el mantenimiento, la escalabilidad y la comprensión del sistema.

## 2. ESPECÍFICOS

- Documentar la estructura modular del software, detallando la funcionalidad de cada componente (carga de datos, simulación de torneos y generación de reportes) para asegurar un entendimiento integral del sistema
- Describir las decisiones de diseño y patrones de programación utilizados, proporcionando directrices claras para futuras modificaciones y mantenimiento del código.

# ESPECIFICACIÓN TÉCNICA

- **REQUISITOS DE HARDWARE**
  - Procesador Mínimo 1 GHz o superior
  - Memoria RAM mínima: 512 MB
  - Al menos 100 MB de espacio disponible para la instalación y ejecución del programa

- **REQUISITOS DE SOFTWARE**
  - Compatible con Windows, Linux y macOS
  - Se puede utilizar cualquier IDE como Eclipse, IntelliJ IDEA o NetBeans para compilar y ejecutar el programa.
  - El software utiliza únicamente las librerías estándar de Java (como java.io, java.util, javax.swing, java.awt, etc.), sin requerir librerías externas adicionales
  - Java Runtime Environment (JRE) o Java Development Kit (JDK) versión 8 o superior

# LÓGICA DEL PROGRAMA

- **Paquete models**

  - **Personaje.java:**
    Define las propiedades y comportamientos de cada personaje:

  - **Atributos**: nombre, salud, ataque, defensa y (calculado) vida inicial.
  - **Métodos:**
    - Métodos getters y setters para cada atributo.
    - Métodos de combate, como calcularDanio(), atacar(), recibirDanio() y estaVivo(), que permiten simular los turnos de ataque y la reducción de vida durante las batallas.

  - **Torneo.java:**
    Encapsula la lógica de la simulación del torneo:
    - Emparejamiento: Itera sobre la lista de personajes para formar parejas.
    - Batallas: Para cada par, se simula un combate por turnos:
      - Cada ataque se ejecuta calculando el daño (ataque del atacante menos defensa del oponente).
      - El combate continúa hasta que uno de los personajes pierde toda su vida.
    - Avance de ronda:
      - Los ganadores de cada batalla se agregan a una lista de sobrevivientes.
      - Si hay un número impar, el personaje sobrante avanza automáticamente.
    - Determinación del campeón: Se repite el proceso hasta que solo queda un personaje, quien es declarado campeón.

- **Paquete utils**
  - **LectorArchivo.java:**
    Se encarga de la lectura y procesamiento del archivo de entrada (.lfp):

    - **Proceso:**
      - Abre el archivo y salta la cabecera.
      - Lee cada línea, separando los datos por el carácter |.
      - Crea objetos Personaje con la información extraída y

los agrega a una lista.
    - **Resultado:** Retorna la lista de personajes que se utilizará en el torneo.

- **ReporteHTML.java:**
  Genera reportes en formato HTML:
    - **Reporte de Mayor Ataque:**
        - Ordena la lista de personajes en orden descendente según el ataque.
        - Extrae el top 5 y genera un archivo HTML con una tabla que muestra el ranking.
    - **Reporte de Mayor Defensa:**
        - Ordena la lista de personajes en orden descendente según la defensa.
        - Extrae el top 5 y genera un archivo HTML similar al de ataque.

➢ **Paquete Principal (lfp.practica)**

- **LFPPractica.java:**
  Actúa como punto de entrada e interfaz de usuario en consola:
    - **Menú Interactivo:**
        - Muestra opciones para cargar el archivo, iniciar el torneo, generar reportes y mostrar información del desarrollador.
    - **Interacción con otros módulos:**
        - Al seleccionar "Cargar archivo", solicita la ruta y llama a LectorArchivo.cargarPersonajes() para obtener la lista de personajes.
        - Al elegir "Jugar (iniciar torneo)", verifica que existan al menos dos personajes y llama a Torneo.iniciarTorneo(), mostrando el desarrollo de las batallas en consola.
        - Las opciones de reporte invocan los métodos de ReporteHTML para generar y, si es posible, abrir los archivos HTML correspondientes.
    - **Control del Flujo:**
        - El menú se repite hasta que el usuario elige la opción "Salir".