Manual técnico:

Introducción:

Este documento detalla el funcionamiento técnico del programa creado con Intellij IDEA (java), desarrollado como parte del curso de **Introducción a la Programación 1**. El sistema permite administrar personajes, registrar peleas entre ellos y consultar información relevante, ejecutándolo completamente en la consola de intellij IDEA.

Está pensado para ser ejecutado en cualquier entorno con Java instalado. Utiliza estructuras básicas como **matrices** para almacenar datos y **bucles** para controlar el flujo del programa.

En este documento se explican:

- Las **variables y estructuras de datos** utilizadas.
- El **funcionamiento de cada opción del menú**.
- Cómo se **validan los datos** ingresados por el usuario.
- La **lógica detrás de las operaciones principales**, como agregar, modificar o eliminar personajes.

Estructura del programa

Método principal: main(String[] args)

Librerías utilizadas:

1. java.util.Scanner: para poder ingresarle datos al programa

2. java.time.LocalDateTime: manejo de fecha.

3. java.time.format.DateTimeFormatter: formateo de fechas

Variables que se utilizaron

variable	tipo	descripción
personaje	String[][]	Matriz que almacena datos de personajes (máx. 100).
totalPersonajes	Int	Contador de personajes registrados.
totalPeleas	int	Contador de peleas registradas.

Sistema do-while

Do: permite mostrar el menú cada que se realice una acción que no sea el “9.” La opción de salir.

While: al elegir la opción de salir “9” muestra un mensaje de despedida y termina el bucle.

Funcionamiento del menú

1. Agregar personaje
    - Se solicita nombre, arma. Habilidades y nivel de poder-
    - Se valida que el nombre no exista y que el nivel este dentro del rango.
    - Se almacena en personajes[totalPersonajes]
2. Modificar personaje.
    - **For** Ejecuta un bloque de código varias veces recorre todos los personajes para buscar uno por nombre.
    - Permite actualizar arma, habilidades y nivel con la variable indexModificar que guarda la posición del personaje en la matriz personajes (sin esta opción el programa no sabría donde están los dados del personaje a modificar)

3. Eliminar personajes.
    - Igualmente que en el modificar personaje en el “case 2” indexModificar busca el personaje que queremos eliminar,
    - solicita confirmación usando if y ! como un operador lógico para confirmar si se desea eliminar el personaje que fue previamente buscado.
4. Ver datos de personajes
    - Si TotalPersonajes es igual a 0, muestra que no hay personajes registrados, si no
    - Muestra el listado de todos lo personajes registrados
    - Solicita el id que va de 0 a 100 de manera que se van registrando personajes.
    - Muestra todos los datos del personaje seleccionado
5. Ver listado de personaje
    - Mostrar nombre nivel de poder de todos los personajes
6. Realizar pelea.
    - Validar que existan 2 personajes
    - Registrar la pelea en historialPeleas con hora y fecha.
7. Ver historial de peleas
    - Muestra todas las peleas registradas con hora y fecha de la pelea apoyado de la variable creada en el case 6: historialPeleas
8. Datos de estudiante.
    - Se muestra la información del estudiante que creo el programa en este caso
      Nombre: Hector Andres Andres Pedro
      Carné: 202307187
      Curso: Laboratorio Introducción a la Programación 1, sección "C"
      Carrera: Ingeniería en Ciencias y Sistemas
      Universidad: San Carlos de Guatemala
9. Salir
    - Muestra mensaje de despedida.

Estructura de datos

personajes= Matriz String[100][8] = almacena los datos de los personajes

historialPeleas= Matriz String[100][3]= guarda fecha y combates realizados

validaciones

Nombre único en case 1: evitar que los nombres estes duplicados

Nivel de poder de (1-100) en case 1 y case 2: el rango permitido del poder de habilidad es de 1 a 100

Existencia de personajes en case 2,3 y 6: evalúa que el exista por lo menos 1 personaje de lo contrario da un mensaje diciendo que no existen personajes o que no se puede realizar la acción