Universidad de San Carlos de Guatemala

Facultad de Ingeniería Escuela de Ciencias y Sistemas

Laboratorio Introducción a la Programación y Computación 1

Sección: D

Segundo Semestre 2025

Ing. Herman Veliz

Aux. Lesther López

Practica 1

Manual técnico

Carlos Didiere Cabrera Rodriguez Carlos Didiere

202401939

INTRODUCCIÓN:

En el siguiente manual se estarán presentando los aspectos técnicos del programa, es decir, todo lo relacionado a la parte programable con el fin de brindar el conocimiento de como las tareas y ejecuciones se están llevando a cabo.

Funcionamiento:

El programa se ejecuta a través de un menú principal que utiliza estructuras de control “do-while" y “switch-case”. Desde este menú, el usuario puede acceder a todas las funcionalidades del sistema de forma ordenada.

El menú principal actúa como la interfaz inicial del proyecto. Su funcionamiento se basa en tres estructuras de control switch-case utilizadas de manera consecutiva, lo que permite presentar un menú en consola de forma organizada y eficiente. Además, esta clase se encarga de invocar los distintos métodos definidos, los cuales analizaremos más adelante en detalle.

Requerimientos:

1. Sistema Operativo: El programa debería poder ejecutarse en la mayoría de los sistemas operativos, incluidos:

    1.2. Windows (7, 8, 10, 11)

    1.3. macOS (10.x o versiones posteriores)

    1.4. Linux (cualquier distribución moderna)

2. Requisitos de Software: Necesitarás el JDK instalado en la máquina. Este es el entorno de desarrollo necesario para compilar y ejecutar programas Java.

    2.1. Versión recomendada: JDK 8 o superior. El programa debería funcionar en versiones más recientes de Java (como JDK 11 o 17) sin problemas, pero en general, JDK 8 es una buena opción de compatibilidad.

3. Un IDE o editor de texto (como IntelliJ IDEA, Eclipse, o incluso VS Code con las extensiones de Java).

Descripción de métodos

1. **Agregar personajes**
   Permite registrar un nuevo personaje en el sistema, almacenando su nombre, arma, habilidades y nivel de poder, siempre que no se haya alcanzado el límite máximo de personajes.

   Parámetros:
   Scanner scanner: Objeto de entrada que permite leer los datos ingresados por el usuario desde la consola.

   Funcionamiento detallado:
   Validación de límite
   Comprueba si el número total de personajes (totalPersonajes) es menor que el máximo permitido (MAX_PERSONAJES = 25).

   Captura de datos del personaje
   Solicita y guarda el nombre del personaje en el arreglo nombres.
   Solicita y guarda el arma del personaje en el arreglo armas.
   Solicita hasta 5 habilidades, almacenándolas en el arreglo habilidadeslista. Si el usuario deja un campo vacío, se detiene el ciclo de captura anticipadamente.
   Solicita el poder del personaje (valor entre 1 y 100) y lo almacena en la posición correspondiente de niveldepoder.

   Actualización del contador
   Incrementa la variable totalPersonajes en 1, indicando que un nuevo personaje ha sido registrado.

2. **Eliminar personajes**
   Permite al usuario eliminar un personaje del sistema, identificándolo por su nombre. Una vez eliminado, todos los datos del personaje (nombre, arma y habilidades) se reorganizan para mantener la estructura del arreglo.

Parámetros:
Scanner scanner: Objeto utilizado para recibir la entrada del usuario desde la consola.

Funcionamiento detallado:
Ingreso del nombre:
Se solicita al usuario ingresar el nombre del personaje que desea eliminar.
El nombre se compara (ignorando mayúsculas/minúsculas) con los datos guardados en el arreglo nombres.

Búsqueda del personaje:
Se recorre el arreglo nombres para buscar coincidencias con el nombre ingresado.
Si se encuentra una coincidencia, se guarda el índice correspondiente.

Eliminación y reorganización:
A partir del índice encontrado, se realiza un desplazamiento de los datos en todos los arreglos relacionados (nombres, armas, habilidades) desde la posición siguiente hasta el final.
Esto permite eliminar los datos del personaje manteniendo el orden de los registros.
Se reduce el contador totalPersonajes en 1.

Manejo de errores:
Si no se encuentra el personaje en el arreglo, se muestra un mensaje de "Personaje no encontrado".

3. **Modificar personajes**
   Este método permite al usuario actualizar los datos de un personaje existente en el sistema. El usuario identifica al personaje por su nombre y luego modifica su nombre, arma y habilidades.

   Parámetros:
   Scanner scanner: Se utiliza para capturar la entrada del usuario desde consola.

   Funcionamiento detallado:
   Entrada del usuario:
   Se solicita al usuario el nombre del personaje a modificar.

Se recorre el arreglo nombres comparando el nombre ingresado (ignorando mayúsculas y minúsculas) para encontrar una coincidencia.

Búsqueda del personaje:
Si se encuentra el personaje, se guarda el índice correspondiente para acceder a sus datos.

Actualización de datos:
Se pide al usuario ingresar el nuevo nombre y se actualiza en nombres.
Se pide una nueva arma y se guarda en armas.
Se solicita una nueva habilidad, la cual se guarda en habilidadeslista.
Confirmación:
Si el personaje se modifica correctamente, se muestra un mensaje indicando que la operación fue exitosa.

Manejo de errores:
Si no se encuentra el personaje, se muestra un mensaje indicando que no existe ningún personaje con ese nombre.

4. **Ver datos de personaje**
   Permite al usuario consultar y visualizar todos los datos registrados de un personaje específico, incluyendo su nombre, arma y lista de habilidades.

   Parámetros:
   Scanner scanner: Utilizado para capturar la entrada del usuario en consola.

   Funcionamiento detallado:
   Solicitud de entrada:
   Se pide al usuario que introduzca el índice del personaje que desea consultar. El rango válido es de 0 hasta totalPersonajes - 1.

   Validación del índice:
   Se verifica que el índice ingresado esté dentro del rango válido.
   Si el índice es incorrecto (fuera de rango), se muestra un mensaje de error.

   Visualización de datos:
   Si el índice es válido, el método imprime:
   Nombre del personaje.

Arma asignada.
Lista de habilidades, una por línea, si están definidas.

5. **Ver listado de personajes**
   Muestra en pantalla una lista con los nombres de todos los personajes registrados actualmente en el sistema, junto con su número de posición.

   Parámetros:
   Este método no recibe parámetros. Utiliza los datos almacenados en los arreglos globales nombres y la variable totalPersonajes.

   Funcionamiento detallado:
   Verificación previa:
   Se verifica si existen personajes registrados (totalPersonajes == 0).
   Si no hay personajes, se muestra el mensaje: "No hay personajes registrados." y se detiene la ejecución del método.

   Recorrido del arreglo:
   Si existen personajes, el método recorre el arreglo nombres[] desde el índice 0 hasta totalPersonajes - 1.

   Impresión ordenada:
   Por cada personaje encontrado, se imprime una línea con el formato:
   n. NombreDelPersonaje,
   donde n representa la posición del personaje (comenzando desde 1).

6. **Realizar pelea**
   Registra una pelea entre dos personajes previamente creados, almacenando un resumen de la pelea en un historial limitado. Este método simula un enfrentamiento sin determinar un ganador.

   Parámetro:
   Scanner scanner: permite capturar desde la consola los índices de los personajes participantes.

   Funcionamiento:
   Verificación de requisitos mínimos:

El método primero comprueba si hay al menos dos personajes registrados en el sistema.
Si no los hay, muestra un mensaje de advertencia y retorna.

Ingreso de personajes:
Solicita al usuario el índice del primer y segundo personaje que participarán en la pelea.
Verifica que los índices estén en el rango válido y que no sean iguales.

Registro de pelea:
Si se cumplen las condiciones y aún hay espacio en el historial de peleas (máximo 25), se guarda la pelea como una cadena de texto en el arreglo historialDePeleas.
Se muestra un mensaje confirmando que la pelea ha sido registrada.

Control de límite de peleas:
Si ya se alcanzó el máximo de peleas permitidas, se muestra un mensaje advirtiendo que no se pueden registrar más.

Variables relacionadas:
nombres: contiene los nombres de los personajes, necesarios para componer el texto descriptivo de la pelea.
historialDePeleas: almacena hasta 25 cadenas de texto representando peleas realizadas.
totalPeleas: contador que indica cuántas peleas han sido registradas.

7. **Ver historial de peleas**

Muestra por consola un listado cronológico de todas las peleas registradas previamente mediante el método `realizarPelea`, permitiendo al usuario visualizar el historial completo.

Parámetros:

Ninguno. El método no recibe argumentos ya que accede directamente a variables globales estáticas del programa.

Funcionamiento:

Verificación inicial:

Comprueba si el contador totalPeleas es igual a cero. En tal caso, significa que no se ha registrado ninguna pelea.

Si no hay peleas, se muestra el mensaje:

No hay peleas registradas.

Recorrido del historial:

Si existen peleas registradas, recorre el arreglo historialDePeleas desde la posición 0 hasta totalPeleas - 1.

Muestra cada entrada numerada en el orden en que se registraron.

Variables relacionadas:

historialDePeleas: Arreglo de tipo String con capacidad máxima de 25 elementos que almacena los detalles de cada pelea.

totalPeleas: Contador que indica cuántas peleas han sido almacenadas hasta el momento.

8. **Ver datos de estudiante:**
   Muestra en consola los datos del estudiante desarrollador del proyecto. Este método sirve para identificar al autor del programa, proporcionando nombre completo y número de carné.

   Parámetros:
   Ninguno.
   No recibe entradas ni interactúa con el usuario; simplemente imprime datos estáticos.

   Funcionamiento:
   Este método está implementado directamente dentro del switch del menú principal, bajo el case 8. Al seleccionarse esta opción:
   Se imprime en pantalla el nombre completo del estudiante.
   Se imprime el número de carné asociado.

No depende de ninguna variable externa ni lógica adicional**.**

9. **Salir del menú:**
   Finaliza la ejecución del programa de manera controlada y limpia, permitiendo al usuario salir del menú principal cuando lo desee.

   Parámetros:
   Ninguno.
   Este bloque no requiere entrada del usuario más allá de la selección de la opción 9 en el menú principal.

   Funcionamiento:
   Cuando el usuario elige la opción 9, el valor de opcion se compara en el switch.
   Se imprime un mensaje de despedida al usuario: "¡Gracias por usar el programa!".
   El valor opcion permanece como 9, lo cual hace que la condición del bucle do-while (while (opcion != 9)) ya no se cumpla.
   El programa sale del bucle y finaliza su ejecución.