

Manual Técnico

1. METODOS CREADOS:

- public static void main(String[] args)

```
public class Ipc1 {  
    public static void main(String[] args)
```

Es un método utilizado que se encuentra al inicio del programa ya que llega a ser el principal. Este método permite que el menú pueda hacer lo que en él se encuentra como agregar personajes, modificarlos, eliminarlos, etc. Además, controla el flujo en el programa y contiene las estructuras switch y if.

- static int buscarPorId(int[] ids, int cantidad, int idBuscado)

```
static int buscarPorId(int[] ids, int cantidad, int idBuscado) {
```

Este método busca por ID dentro del arreglo ids y también devuelve posición

2. ESTRUCTURA DEL PROGRAMA:

- Paquetes de librerías importadas

```
import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Scanner;
```

Estas librerías importadas son muy importantes, la primera ayuda a obtener la hora y fecha actual.

La segunda ayuda a formatear las fechas y esta muy ligada con la anterior por ello es muy importante ponerlas juntas.

Y la ultima ayuda a capturar la entrada que el usuario proporciona.

- `public static void main(String[] args) {`

Acá se define las variables, el menú principal y la lógica que lleva el código.

Además, se declaró que únicamente se iban a trabajar con 100 personajes por lo que en el menú y todas sus opciones solamente se van a aceptar 100 entradas.

Acá se puede observar como es que el programa solo aceptará 100 datos de entrada que será el limite de personajes permitidos, así mismo será con las habilidades, las peleas y su historial.

- Menú

```
while (opcion != 9) {  
    System.out.println("\n---- MENU ----");  
    System.out.println("1. Agregar personajes");  
    System.out.println("2. Modificar datos de personajes");  
    System.out.println("3. Eliminar personajes");  
    System.out.println("4. Ver datos de un personaje");  
    System.out.println("5. Ver datos de personajes");  
    System.out.println("6. Realizar pelea entre personajes");  
    System.out.println("7. Ver historial de peleas");  
    System.out.println("8. Ver datos del estudiante");  
    System.out.println("9. Salir");  
    System.out.print("Elige una opcion: ");  
    opcion = scanner.nextInt();  
    scanner.nextLine();  
}
```

El while es un ciclo que está en el método main, este ciclo se encarga de mostrar el menú al usuario, esperar que el usuario marque la opción que desea y luego continuar según lo que se haya colocado, por ejemplo, en este código se hizo del 1 al 9 y se le pregunta al usuario que desea realizar.

Si el usuario coloca un numero diferente de 1 al 9 el menú se volverá a mostrar.

- Switch

```
-----  
  
switch (opcion) {  
    case 1:
```

El switch es muy importante ya que este sirve para ejecutar un bloque de código que va a depender de lo que el usuario coloque dentro del while que será el menú, si el usuario coloca mal algo el switch como está conectado con el while pasara a preguntar que opción desea nuevamente.

Este programa contiene dentro del switch 9 opciones diferentes para que el usuario puede colocar lo que desee ver.

- Case

```
case
```

El case por ir dentro del switch representa una posible opción que el usuario pueda colocar que en este caso son 9 case.

- Case 1

Si el usuario coloca 1 en el menú se ejecutará esto, en donde preguntará el nombre del personaje.

- If

Este es una condición, en el caso del case 1 si el usuario no coloca nada en agregar personajes se mostrará “El nombre no puede estar vacío” de lo contrario seguirá ejecutándose todo con normalidad.

- Boolean y for

```
boolean existe = false;
```

El boolean contiene 2 valores que son falso y verdadero, estos permiten evaluar condiciones, como por ejemplo en el código si el usuario ingresa un nombre que ya existe el boolean pasara a decir que es verdadero o falso y si es verdadero mostrara el mensaje “Ese nombre ya existe”, de lo contrario agregara de forma correcta el nombre ingresado.

El for es una estructura de repetición que ejecuta un número determinado de veces.

- Todo lo demás del case 1/ arma del personaje

```
System.out.print("Arma del personaje: ");
String arma = scanner.nextLine().trim();

if (arma.isEmpty()) {
    System.out.println("El arma no puede estar vacia.");
    break;
```

Acá se le pide al usuario que coloque el nombre de un arma, pero también el código revisa mediante un if si el usuario no colca nada le mostrará el mensaje “El arma no puede estar vacía” y así servirá para los demás case.

- Habilidades

Acá se muestra un mensaje de ingresar 5 habilidades, y por pedirle 5 al usuario se deberá hacer un conteo y es lo que hace el int, y así mismo si el usuario no coloca nada se ejecutara el if y se mostrara el mensaje “la habilidad no puede estar vacía”

- Nivel de poder

En el nivel que se le pedirá al usuario de nivel será de 1 a 100 por el cual el if revisara si lo ingresado si esta entre ese parámetro o sino no, y el else revisara si el usuario ingreso un número o si ingreso algo más, si ese es el caso mostrara el mensaje “Error, debe ser un número”

- Guardar personaje

```
nombres[cantidad] = nombre;
armas[cantidad] = arma;
fuerzas[cantidad] = fuerza;
ids[cantidad] = idCounter++;
cantidad++;

System.out.println("Personaje agregado con exito.");
break;
```

Para terminar el case 1 esta esta función la cual sirve para guardar todo lo pedido anterior y mostrar el mensaje “Personaje agregado con éxito”, además se guardará cada personaje con un ID diferente que empieza desde el 1 al 100.

- Case 2

El case 2 servirá para modificar el personaje que anteriormente se colocó, mediante el ingreso de su ID o del nombre del personaje guardado.

- Case 3

El case 3 se eliminará el personaje según el ID proporcionado por el usuario, pero también habrá un if el cual buscare si el ID se encuentra de lo contrario mostrara el mensaje “Personaje no encontrado” y se eliminara esa posición.

- Case 4

En el case 4 se consultará la información de un solo personaje lo cual con su ID se podrá identificar de que personaje se esta preguntando y mostrara toda la información de ese personaje.

- Case 5

```
case 5:
    if (cantidad == 0) {
        System.out.println("No hay personajes registrados en el sistema.");
    } else {
        System.out.println("----- LISTADO DE PERSONAJES -----");
        for (int i = 0; i < cantidad; i++) {
            System.out.println("ID: " + ids[i] +
                               " | Nombre: " + nombres[i] +
                               " | Nivel de poder: " + fuerzas[i]);
        }
        break;
    }
}
```

En el case 5 servirá para mostrar los nombres y toda la información de todos los personajes ingresados anteriormente.

- Case 6

```

        case 6:
            if (cantidad < 2) {
                System.out.println("Se necesitan al menos 2 personajes para una pelea.");
                break;
            }

            System.out.print("ID del primer personaje: ");
            int id1 = scanner.nextInt();
            System.out.print("ID del segundo personaje: ");
            int id2 = scanner.nextInt();
            scanner.nextLine();

            int i1 = buscarPorId(ids, cantidad, id1);
            int i2 = buscarPorId(ids, cantidad, id2);

            if (i1 == -1 || i2 == -1) {
                System.out.println("Uno o ambos personajes no existen.");
                break;
            }

            if (i1 == i2) {
                System.out.println("No puedes enfrentar a un personaje consigo mismo.");
                break;
            }

            String ganador;
            if (fuerzas[i1] > fuerzas[i2]) {
                ganador = nombres[i1];
            } else if (fuerzas[i2] > fuerzas[i1]) {
                ganador = nombres[i2];
            } else {
                ganador = "Empate";
            }

            LocalDateTime fechaHora = LocalDateTime.now();
            DateTimeFormatter formato = DateTimeFormatter.ofPattern("dd/MM/yyyy HH:mm:ss");
            String fechaFormateada = fechaHora.format(formato);

            String pelea = nombres[i1] + " (" + fuerzas[i1] + ") vs " +
                nombres[i2] + " (" + fuerzas[i2] + ") -> Ganador: " + ganador +
                " | Fecha: " + fechaFormateada;

            historialPeleas[peleasRealizadas++] = pelea;

            System.out.println("--- Resultado de la pelea ---");
            System.out.println(pelea);
            break;

```

En el case 6 se realizará la pelea, cuando el usuario ingreso los datos de cada personaje también se le pidió fuerza y acá va a servir para las batallas, el personaje con mayor fuerza ganará

- Case 7

```
case 7:
    if (peleasRealizadas == 0) {
        System.out.println("No hay peleas registradas en el historial.");
    } else {
        System.out.println("--- Historial de Peleas ---");
        for (int i = 0; i < peleasRealizadas; i++) {
            System.out.println((i + 1) + ". " + historialPeleas[i]);
        }
    }
}
```

En el case 7 se revisarán las batallas

- Case 8 y 9

En el case 8 simplemente se muestran los datos de mi persona mediante el `system.out.println` y el `break` cierra el case

Y en el case 9 que es la última opción del menú se finaliza el programa exitosamente, pero si el usuario ingresa otra cosa que no sea 1-9 se mostrará el mensaje “Opción no valida”, y le mostrara nuevamente el menú mediante el `while`.