



Universidad de San Carlo de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Introducción a la programación y computación 1
Ing. Moisés Velásquez
Auxiliar. Pablo Oliva

PRACTICA No.1: INFORME DE DESARROLLO

Guillermo Enrique Marroquin Morán

202103527

Guatemala 15 de ago. de 25

Indice

Tabla de contenido

<i>Indice</i>	2
<i>Informe de desarrollo de la práctica 1</i>	3
¿Cómo desarrolle la práctica?	3
¿Qué problemas tuve con la práctica?	6
¿Cómo se pudo solucionar los problemas.....	7
Menú interactivo.....	8
<i>EGRAFIA</i>	13

Informe de desarrollo de la práctica 1

¿Cómo desarrolle la práctica?

Esta práctica se llevó a cabo con el lenguaje de java con el IDE de NetBeans, parte de esta tuvo un registro en git, subiéndose *commits* al repositorio en GitHub, primeramente, hice lectura de esta anotando los puntos clave en la aplicación de notas para una comprensión un poco más resumida de la misma. Primeramente, hice un diagrama de flujo sencillito para que me pudiera guiar a modo de cómo debería ir al programa en términos generales.

Luego de eso empecé a desarrollar lo que sería el menú principal usado el ciclo “do-while” para que ejecutara cada opción, hasta que se eligiese la novena opción de este que finalizaría el programa

```
do{
    // Deplegar el Menu principal
    System.out.println("Menu principal");
    System.out.println("1. Agregar Personaje");
    System.out.println("2. Modificar Personaje ");
    System.out.println("3. Eliminar Personaje");
    System.out.println("4. Ver Datos de un Personaje");
    System.out.println("5. Ver Listado de Personaje");
    System.out.println("6. Realizar pelea entre Personajes");
    System.out.println("7. Ver Historial de peleas");
    System.out.println("8. Ver datos de Estudiante");
    System.out.println("9. Salir");
    System.out.print("Elige una opción: ");
} while (escribir != 9); // Todo esto se repite indefinidamente hasta que pongan 9
scanner.close();
}
```

Seguido de eso se desarrollaron la primera y octava opción, que son el de agregar y ver mis datos, inicialmente se habían usado arraylist para eso, aunque luego por penalizaciones se tuvo que eliminar esa función y usar una matriz de 25*8 a modo de una forma de “celdas de memoria”, para almacenar los datos de los personajes, y se implementó un arreglo para las 5 habilidades, de los personajes, como dato inicialmente quería que el nivel de poder tuviese más relevancia, pero al final lo deje como un valor numérico sin más.

```
public static void main(String[] args){
    Scanner scanner = new Scanner(System.in);

    // Declaramos las variables
    int escribir;
    String [][] personajes = new String [25][8]; // Matriz de personajes
    int cantidadPer = 0; //esto es el contador de personajes registrados

    String [] historialCo = new String [50]; //para guardar en matriz hasta 50 madrazos
    int cantPeleas = 0;
    String continuar;

    Random random = new Random();
```

```
escribir = scanner.nextInt();
//Ejecutar segun las opciones que se ponga en la consola, pudo haber sido if's, pero es mas practico switch
switch (escribir) {
    case 1:
        do{
            if (cantidadPer >= 25){
                System.out.println("Ya llegaste al maximo numero de personajes registrados");
                break;
            }
            scanner.nextLine();
            System.out.println("Registro de un nuevo personaje");

            //verificamos si el nombre esta repetido o no
            String nombre;
            boolean NoRepetido = false;
            do{
                System.out.println("Ingresa el nombre del personaje");
                nombre = scanner.nextLine();
                for(int i = 0; i < cantidadPer; i++){
                    //el equals es para "ignorar" mayusculas y minusculas del nombre en cuestion
                    if(personajes[i][0] != null && personajes[i][0].equalsIgnoreCase(nombre)){
                        NoRepetido = true;
                        System.out.println("Ese nombre ya fue registrado, elige otro nombre");
                        break;
                    }
                }
            } while (NoRepetido);

            System.out.println("Ingresa el arma de tu preferencia");
            String arma = scanner.nextLine();
            int nivel_podi=0;
```

Para las opciones de modificar y eliminar personaje hice dos módulos uno para buscar personajes, en la matriz y uno para mostrar dichos personajes, el primer módulo mencionado lo que hace es buscar el nombre usando el bucle for, empezando en 0 hasta el -1, comprobando que en la fila algún nombre no este vacío y buscando por nombre sin importar las mayúsculas o minúsculas esto gracias a "equalsIgnoreCase", el segundo modulo lo que hace es mostrar en la consola los personajes en forma de lista, aunque estos esten almacenados en una matriz, cabe recalcar que para las habilidades, al ser tecnicamente independientes van desde la posicion 3 hasta la 8 esto por el ciclo for.

```
// el modulo de buscar personajes, este se usa en las opciones 2,3,4,5,6 y 7, waw que monton
public static int buscarPer (String [][] personajes, int cantidad, String nombre){
    for (int i = 0; i < cantidad; i++){
        if (personajes[i][0] != null && personajes[i][0].equalsIgnoreCase(nombre)){
            return i;
        }
    }
    return -1; // esto indica que si el personaje no se encuentra va a retornar un -1
}

//el modulo de mostrar personaje, se usa en las opciones 2,4,5 y posiblemente 7
public static void mostrarPer (String [][] personajes, int Per){
    System.out.println("Personaje");
    System.out.println("nombre:" + personajes[Per][0]);
    System.out.println("arma:" + personajes[Per][1]);
    System.out.println("nivel de poder:" + personajes[Per][2]);
    System.out.println("habilidades:");
    for (int j = 3; j < 8; j++){
        System.out.println("nombre:" + personajes[Per][j]);
    }
}
```

Para las opciones 2,3,4,5,6,7 lo que hace el modulo de buscar personajes, al no encontrarlos retornara un -1 esto con la condicion if, hace que este mande el mensaje que el personaje solicitado no existe.

```
Menu principal
1. Agregar Personaje
2. Modificar Personaje
3. Eliminar Personaje
4. Ver Datos de un Personaje
5. Ver Listado de Personaje
6. Realizar pelea entre Personajes
7. Ver Historial de peleas
8. Ver datos de Estudiante
9. Salir
Elige una opción: 2
Modificar atributos del Personaje
Ingresa el nombre del personaje a quien quieras modificar
mario
Ese personaje no existe, ingresa otro nombre
```

Siguioendo con el desarrollo las opciones de ver atributos de un personaje y ver listado de personajes son casi lo mismo solo con la diferencia de que en el de ver atributos del personaje se usar el modulo de buscar personaje y muestra, los datos de ese personaje en concreto, y la otra opcion solo muestra el listado de todos los personajes ingresados y ya.

```

case 4:
    System.out.println("Ver atributos del personaje:");
    scanner.nextLine();

    System.out.println("Ingresa el nombre del personaje del que quieras ver sus atributos");
    String nombreBus = scanner.nextLine();

    int Buscar = buscarPer (personajes, cantidadPer, nombreBus);
    if (Buscar == -1){
        System.out.println("No se encontro al personaje, deberias intentar con otro nombre");
    } else {
        mostrarPer (personajes, Buscar);
    }
    break;

//Hacemos un llamado del modulo de mostrar personajes y del modulo de buscar
case 5:
    System.out.println("Listado de de personajes");
    if(cantidadPer == 0){
        System.out.println("No hay ningun personaje registrado... de momento");
    } else{
        for (int i = 0; i < cantidadPer; i++){
            mostrarPer (personajes, i); // llamado del modulo de mostrar personajes
            System.out.println();
        }
    }
    break;

```

Para las ultimas opciones que son las de realizar combate entre personajes y de ver el registro de las mismas, de igual manera se hace llamado del modulo de buscar personajes, pero este al buscar dos personajes, imprime un mensaje en pantalla, que marca si uno o los dos personajes no existen como tal, luego con la librería “random” se elige al azar entre 0 o 1 si el 0 le toca al primer personaje o al segundo este gana, y se registra la victoria del mismo, con la fecha y hora actuales del ordenador en formato (dia/mes/año, hora/minuto/segundo), esto gracias a las librerías, *time.LocalDateTime* y *time.format.DateFormatter*

```

case 6:
    System.out.println("Realizar combate entre personajes");
    scanner.nextLine();

    System.out.println("Ingresa el nombre del primer peleador");
    String per1 = scanner.nextLine();
    int ID1 = buscarPer (personajes, cantidadPer, per1);

    System.out.println("Ingresa al segundo peleador");
    String per2 = scanner.nextLine();
    int ID2 = buscarPer (personajes, cantidadPer, per2);

    if(ID1== -1 || ID2== -1){
        System.out.println("Un personaje o los dos no existen, todavia");
    } else{
        // Usare la libreria random porque quiero y puedo
        int ganador = random.nextInt(2); //0 para un personaje 1 para el otro
        String resultado;
        if (ganador == 0){
            resultado = personajes [ID1][0] + "pierdes contra" + personajes[ID2][0];
        } else{
            resultado = personajes [ID2][0] + "pierdes contra" + personajes[ID1][0];
        }
        String fecha_hora = LocalDateTime.now ().format(DateTimeFormatter.ofPattern ("dd/MM/yyyy HH:mm:ss"));
        String Registrar = fecha_hora + "---" + resultado;

        if (cantPeleas < historialCo.length){
            historialCo[cantPeleas] = Registrar;
            cantPeleas++;
        }
        System.out.println("Combate registrado:" + Registrar);
    }
    break;

```

```

case 7:
    System.out.println("Ver historial de combates");
    if(cantPeleas == 0){
        System.out.println("No hay peleas registradas... aun");
    }else {
        for(int i=0; i<cantPeleas; i++){
            System.out.println(historialCo[i]);
        }
    }
    break;

```

Como opcion final, son ver mis datos, que es mi nombre completo, mi carnet universitario y solamente, y la ultima opcion que es salir.

```

case 8:
    System.out.println("Este programa fue hecho por Guillermo Enrique Marroquin Morán");
    System.out.println("202103527");
    System.out.println("Este programa tiene derechos de autor o no...");
    break;
case 9:
    System.out.println("BYE BYEEEEEE");
    break;
default:
    System.out.println("!Esa opcion no existe!, ingresa otra opción valida.");
}
} while (escribir != 9); // Todo esto se repite indefinidamente hasta que pongan 9
scanner.close();
}
}

```

¿Qué problemas tuve con la práctica?

Tuve dificultades a la hora de crear el repositorio de la práctica dado que, en MacOS, no se puede copiar la ruta de una carpeta como tal, sino que hay que estar en la terminal de comandos aplicando los comandos "cd" y "ls" para abrir las carpetas para subir el repositorio.

A la hora de programar una matriz para que se guardasen los personajes, inicialmente tenía la idea que esta fuese variable, con esto me refiero a que se declarasen constantes y hacer la matriz de personajes con los 8 atributos (nombre, arma, nivel de poder y las 5 habilidades) que esta fila fuese constante porque no piden más que solo eso, y las columnas fuesen variables, que podrían modificarse fácilmente con cambiar el valor de la constante y la restricción de la misma, pero al final hacer eso el programa no quería correr, y se quedaba cargando indefinidamente.

Inicialmente como no sabía las penalizaciones, use arraylist en el código, con eso vi un video en YouTube de constructores, de allí hice la clase personajes, de allí luego de preguntar si se podía o no, me dijeron que no, con lo que elimine el uso de arraylist, pero la clase de personajes al eliminarla no ejecuta el programa, incluso al copiar y pegar las variables declaradas, seguía sin ejecutarse.

A la hora de poner las líneas de comando para escribir las 5 habilidades, el programa me detectaba que la variable no había sido encontrada, y yo la había declarado un arreglo en una clase por aparte llamada "personajes" y no quería funcionar.

A la hora de escribir el nivel de poder, el programa se quedaba cargando, sin querer avanzar, se quedaba congelado y no quería correr, esto porque está implementando un bucle "do-while", para que solo dejase ingresar números en el nivel de poder, pero al ponerla pasaba todo lo que dije anteriormente.

Por ultimo y no más importante, cuando se realizaban los combates entre personajes, había implementado que fuese random, pero el nivel de poder influyese un poco en la victoria o derrota de un personaje, pero a la hora de implementarlo, no funcionaba y el programa no se ejecutaba.

¿Cómo se pudo solucionar los problemas

Respecto al repositorio, con la actividad número 3 pude familiarizarme un poco, con la instalación y los comando de git, para así poder crear un repositorio desde la terminal de control y no solo arrastrar los archivos de la práctica.

Para solucionar el dilema de la matriz, llegue a una rápida conclusión, crear una matriz fija de 25 columnas y 8 filas, esto para que se guarden los personajes como si fuera celdas de una memoria, y a la hora de mostrar los personajes, se muestre como una lista.

Con el problema de la clase personajes se quedó en la carpeta de la práctica, aunque logre solucionar que si se elimina eso no pasa nada, pero allí se queda.

El problema de que no podía encontrar el arreglo de las 5 habilidades, solo se me ocurrió copiar de la clase de personajes la línea: `"String [] habilidades = new String [5];"` ponerlo antes de del ciclo "for" y a la hora de imprimir el número de habilidad puse que fuese "i+1" para que imprima la habilidad(i+1).

Lo único que se me ocurrió para el problema de que se quedase cargando sin avanzar fue dejar que el nivel de poder fuese un valor cualquiera sin más, eliminando el bucle "do-while", que restringía el uso exclusivamente de números al nivel de poder.

Como en el inciso anterior hice mención el nivel de poder solo sería un valor numérico sin más, por lo que descarte cualquier interacción de este con el resultado de realizar un combate entre personajes, por lo tanto, solo hice que los combates sea 100% aleatorios.

Menú interactivo

Primera opción para elegir que es la de registrar un nuevo personaje, y la pregunta que si quieres agregar a otros, como máximo 25.

```
Menu principal
1. Agregar Personaje
2. Modificar Personaje
3. Eliminar Personaje
4. Ver Datos de un Personaje
5. Ver Listado de Personaje
6. Realizar pelea entre Personajes
7. Ver Historial de peleas
8. Ver datos de Estudiante
9. Salir
Elige una opción: 1
Registro de un nuevo personaje
Ingrese el nombre del personaje
Megaman
Ingresa el arma de tu preferencia
Megabuster
Ingresa el nivel de poder (de 0 a 100)
100
Habilidad #1: Charge shot
Habilidad #2: Slide
Habilidad #3: Top spin
Habilidad #4: rush coil
Habilidad #5: Energy tank
¿Deseas ingresar otro personaje? (s/n):
```

Después de elegir “s” (ojo dar doble enter) en el menú de si quieres ingresar a otro personaje, puedes agregar un nuevo personaje, hasta que elijas “n” o llegues a más de 25 personajes.

```
¿Deseas ingresar otro personaje? (s/n): s
```

```
Registro de un nuevo personaje
Ingrese el nombre del personaje
Kirby
Ingresa el arma de tu preferencia
Star road
Ingresa el nivel de poder (de 0 a 100)
100
Habilidad #1: copiar
Habilidad #2: Aliento de fuego
Habilidad #3: estrella remolque
Habilidad #4: el poder de la amistad
Habilidad #5: rayo de espada
¿Deseas ingresar otro personaje? (s/n): |
```


Al escribir la "n", volverás al menú principal.

¿Deseas ingresar otro personaje? (s/n): n

Menu principal

1. Agregar Personaje
2. Modificar Personaje
3. Eliminar Personaje
4. Ver Datos de un Personaje
5. Ver Listado de Personaje
6. Realizar pelea entre Personajes
7. Ver Historial de peleas
8. Ver datos de Estudiante
9. Salir

Al elegir la segunda opción puedes modificar los atributos de un personaje

Elige una opción: 2

Modificar atributos del Personaje

Ingresar el nombre del personaje a quien quieras modificar

Varita estelar

Estos son los datos actuales del personaje

Personaje

nombre:Varita estelar

arma:Kirby

nivel de poder:0

habilidades:

nombre:Copiar

nombre:Aliento de fuego

nombre:estrella remolque

nombre:El poder de la amistad

nombre:Rayo de espada

Ingresar un nuevo nombre

Mario

Elige un arma nueva

Guantes de almidon

Ingresar un nuevo nivel de poder

50

Nueva habilidad #1:

Super salto

Nueva habilidad #2:

Salto en la pared

Nueva habilidad #3:

Bola de fuego

Nueva habilidad #4:

Estrella de poder

Nueva habilidad #5:

El mario tornado

Luego de eso regresas al menú principal de nuevo

Menu principal

1. Agregar Personaje
2. Modificar Personaje
3. Eliminar Personaje
4. Ver Datos de un Personaje
5. Ver Listado de Personaje
6. Realizar pelea entre Personajes
7. Ver Historial de peleas
8. Ver datos de Estudiante
9. Salir

Elige una opción: |

Al elegir la tercera opción puedes elegir eliminar el personaje que escribas su nombre.

Elige una opción: 3

Elimina un personaje

Pon el nombre del personaje que quieras eliminar

Megaman

El personaje se ha eliminado

Al elegir la cuarta opción se te pedirá escribir el nombre del personaje que quieras ver los datos.

Elige una opción: 4

Ver atributos del personaje :

ingresa el nombre del personaje del que quieras ver sus atributos

Mario

Personaje

nombre:Mario

arma:Guantes de almodin

nivel de poder:0

habilidades:

nombre:Super Salto

nombre:salto de pared

nombre:Bola de fuego

nombre:Pisoton

nombre:El mario tornado

Al elegir la opción número 5 aparecerán los datos de los personajes actuales.

Elige una opción: 5
Listado de de personajes
Personaje
nombre:Mario
arma:Guantes de almodin
nivel de poder:0
habilidades:
nombre:Super Salto
nombre:salto de pared
nombre:Bola de fuego
nombre:Pisoton
nombre:El mario tornado

Personaje
nombre:Kirby
arma:Varita estelar
nivel de poder:0
habilidades:
nombre:Copiar
nombre:Aliento de fuego
nombre:estrella remolque
nombre:el poder de la amista
nombre:rayo de espada

Al elegir la sexta opción se los personajes “pelearan”, y se mostrara el ganador, junto a la hora y fecha actual.

Elige una opción: 6
¡Realizar combate entre personajes!
Ingresa el nombre del primer peleador
Mario
Ingresa al segundo peleador
Kirby
Combate registrado:15/08/2025 18:53:05---Mariopierdes contraKirby

Al elegir la séptima opción, se mostrarán los combates realizados junto a su debida hora y fecha.

```
Elige una opción: 7
Ver historial de combates
15/08/2025 18:53:05---Mariopierdes contraKirby
15/08/2025 18:54:54---Mariopierdes contraKirby
.....
```

Al elegir la octava opción se mostrarán mis datos.

```
Elige una opción: 8
Este programa fue hecho por Guillermo Enrique Marroquin Morán
202103527
Este programa tiene derechos de autor o no...
```

Al elegir la novena opción se finaliza el programa.

```
Elige una opción: 9
BYE BYEEEEE
BUILD SUCCESSFUL (total time: 14 minutes 36 seconds)
```

EGRAFIA

- Squirrels, J. (2023, February 14). *Índice de cadena Java de ()*. CodeGym.
Retrieved August 15, 2025,
from <https://codegym.cc/es/groups/posts/es.1064.ndice-de-cadena-java-de->
- H, T. (2020, September 17). *Java - limit number between min and max*. Stack
Overflow. Retrieved August 15, 2025, from <https://stackoverflow-com.translate.goog/questions/17933493/java-limit-number-between-min-and-max? x tr sl=en& x tr tl=es& x tr hl=es& x tr pto=tc>
- Pabooliva. (n.d.). *GitHub - 27Pabooliva27/IPC1C2S2025*. GitHub. Retrieved
August 15, 2025, from <https://github.com/27Pabooliva27/IPC1C2S2025.git>
- Aprende a Programar. (2024, November 4). 🧑‍🎓 ¿Qué es un constructor en Java? |
POO con Java 🖥️ | *Curso Maestro de JAVA* 🔥 *Episodio #27* [Video].
YouTube. Retrieved August 15, 2025,
from <https://www.youtube.com/watch?v=bJzxAJVF-PI>
- InformativaC. (2018, December 3). *Eliminar objetos en un vector / Java* [Video].
YouTube. Retrieved August 15, 2025,
from <https://www.youtube.com/watch?v=LgD5mBHO6fQ>
- González, J. D. M. (2020, April 26). *Final y constantes*. Retrieved August 15, 2025,
from <https://www.programarya.com/Cursos/Java/Sistema-de-Tipos/Final-y-Constantes>
- González, J. D. M. (2020b, April 26). *Final y constantes*. Programar Ya. Retrieved
August 15, 2025, from <https://www.programarya.com/Cursos/Java/Sistema-de-Tipos/Final-y-Constantes>
- Mendez, O. (2022, February 19). *¿como cambiar las letras mayusculas a minusculas y viceversa?* Stack Overflow En Español. Retrieved August 15, 2025, from <https://es.stackoverflow.com/questions/515924/como-cambiar-las-letras-mayusculas-a-minusculas-y-viceversa>

Obregon, A. (2024, July 24). *Java's String.valueOf() Method Explained*. Medium.

Retrieved August 15, 2025, from https://medium-com.translate.goog/@AlexanderObregon/javas-string-valueof-method-explained-b3fba964d3ec? x tr sl=en& x tr tl=es& x tr hl=es& x tr pto=sge& x tr _hist=true

Hayes, H. (2012, season-02). *One moment, please. . .* Beginners Book. Retrieved

August 15, 2025, from <https://beginnersbook-com.translate.goog/2017/10/java-string-valueof-method/? x tr sl=en& x tr tl=es& x tr hl=es& x tr pto=sge>

Squirrels, J. (2023b, February 14). *Método String equalsIgnoreCase() en Java*.

CodeGym. Retrieved August 15, 2025, from <https://codegym.cc/es/groups/posts/es.816.mtodo-string-equalsignorecase-en-java>

parseInt() - JavaScript | MDN. (2025, June 24). MDN Web Docs. Retrieved August 15, 2025,

from https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/parseInt

Berenguer, M. C., & Berenguer, M. C. (2018, February 27). Fechas en Java:

LocalDate, LocalTime y LocalDateTime. *JavAutodidacta*. Retrieved August 15, 2025, from <https://javautodidacta.es/tiempo-en-java-localdate-localtime/#tab-con-7>