

Informe de Laboratorio 01

Tema: Laboratorio 01

Nota

Estudiante	Escuela	Asignatura
Victor Mamani Anahua vmamanian@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación II Semestre: II Código: 20230489

Laboratorio	Tema	Duración
01	Laboratorio 01	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 10 Setiembre 2023	Al 17 Setiembre 2023

1. Tarea

- Escribir un programa donde se creen 5 soldados considerando sólo su nombre. Ingresar sus datos y después mostrarlos. Restricción: se realizará considerando sólo los conocimientos que se tienen de FP1 y sin utilizar arreglos estándar, sólo usar variables simples.
- Escribir un programa donde se creen 5 soldados considerando su nombre y nivel de vida. Ingresar sus datos y después mostrarlos. Restricción: se realizará considerando sólo los conocimientos que se tienen de FP1 y sin utilizar arreglos estándar, sólo usar variables simples.
- Escribir un programa donde se creen 5 soldados considerando sólo su nombre. Ingresar sus datos y después mostrarlos. Restricción: aplicar arreglos estándar.
- Escribir un programa donde se creen 5 soldados considerando su nombre y nivel de vida. Ingresar sus datos y después mostrarlos. Restricción: aplicar arreglos estándar. (Todavía no aplicar arreglo de objetos).
- Escribir un programa donde se creen 2 ejércitos, cada uno con un número aleatorio de soldados entre 1 y 5, considerando sólo su nombre. Sus datos se inicializan automáticamente con nombres tales como "Soldado0", "Soldado1", etc. Luego de crear los 2 ejércitos se deben mostrar los datos de todos los soldados de ambos ejércitos e indicar qué ejército fue el ganador. Restricción: aplicar arreglos estándar y métodos para inicializar los ejércitos, mostrar ejército y mostrar ejército ganador. La métrica a aplicar para indicar el ganador es el mayor número de soldados de cada ejército, puede haber empates. (Todavía no aplicar arreglo de objetos)

2. Equipos, materiales y temas utilizados

- Sistema Operativo Ubuntu GNU Linux 23 lunar 64 bits Kernell 6.2.v
- Sistema Operativo Windows 11.
- Visual Studio Code.
- OpenJDK 64-Bits 19.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.
- Programación Orientada a Objetos.
- Actividades del Laboratorio 01

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/VictorMA18/fp2-23b.git>
- URL para el laboratorio 01 en el Repositorio GitHub.
- <https://github.com/VictorMA18/fp2-23b/tree/main/Fase01/Lab01>

4. Actividades del Laboratorio 01

4.1. Ejercicio 01

- En el primer ejercicio en el comit especificamos y aplicamos un for para el ingreso de datos que seria el nombre de los soldados.
- El codigo , la ejecucion y el commit seria el siguiente:

Listing 1: Commit

```
$ git commit -m "Creamos el 1er ejercicio donde creamos 5 soldados y imprimimos sus  
datos"
```

Listing 2: Las lineas de codigos del Ejercicio01 serian

```
// Laboratorio Nro 1 - Ejercicio1  
// Autor: Mamani Anahua Victor Narciso  
// Colaboro:  
// Tiempo:  
import java.util.*;  
public class Videojuego{  
    public static void main(String args[]){  
        //1. DECLARAMOS LAS VARIABLES  
        Scanner sc = new Scanner(System.in);  
        String name;  
        //2. LE PEDIMOS A LOS 5 USUARIOS ESCRIBIR SUS NOMBRES  
        for(int i = 0; i < 5; i++){
```

```
        System.out.println("Ingrese su nombre: ");
        name = sc.nextLine();
        System.out.println("El nombre del soldado " + (i + 1) + " es: " + name);
    }
}
```

Listing 3: La ejecucion dada:

```
Ingrese su nombre:
JUAN
El nombre del soldado 1 es: JUAN
Ingrese su nombre:
BETO
El nombre del soldado 2 es: BETO
Ingrese su nombre:
TITO
El nombre del soldado 3 es: TITO
Ingrese su nombre:
HERNAN
El nombre del soldado 4 es: HERNAN
Ingrese su nombre:
HECTOR
El nombre del soldado 5 es: HECTOR
```

4.2. Ejercicio 02

- En el segundo ejercicio en el comit especificamos y aplicamos dos for para el ingreso de datos que seria el nombre de los soldados y vida de los soldados.
- El codigo , la ejecucion y el commit seria el siguiente:

Listing 4: Commit

```
$ git commit -m "Creamos el 2do ejercicio donde creamos 5 soldados y imprimimos sus
nombres y su vida y tambien borramos el archivo Sumadig.java ya que no lo
necesitamos"
```

Listing 5: Las lineas de codigo del Ejercicio02 serian

```
// Laboratorio Nro 1 - Ejercicio2
// Autor: Mamani Anahua Victor Narciso
// Colaboro:
// Tiempo:
import java.util.*;
public class Videojuego{
    public static void main(String args[]){
        //1. DECLARAMOS LAS VARIABLES
        Scanner sc = new Scanner(System.in);
        String name;
        int health;
        //2. LE PEDIMOS A LOS 5 USUARIOS ESCRIBIR SUS NOMBRES Y SU NIVEL DE VIDA
        for(int i = 0; i < 5; i++){
            System.out.println("Ingrese su nombre: ");
```

```
        name = sc.next();
        System.out.print("El nombre del soldado " + (i + 1) + " es: " + name);
        System.out.println("");
        for(int x = 0; x < 1; x++){
            System.out.println("Ingrese su nivel de vida: ");
            health = sc.nextInt();
            System.out.println("El nivel de vida del soldado " + (i + 1) + " es: " +
                               health);
        }
    }
}
```

Listing 6: La ejecución dada:

```
Ingrese su nombre:
JUAN
El nombre del soldado 1 es: JUAN
Ingrese su nombre:
BETO
El nombre del soldado 2 es: BETO
Ingrese su nombre:
TITO
El nombre del soldado 3 es: TITO
Ingrese su nombre:
HERNAN
El nombre del soldado 4 es: HERNAN
Ingrese su nombre:
HECTOR
El nombre del soldado 5 es: HECTOR
```

4.3. Ejercicio 03

- En el tercer ejercicio en el commit especificamos y aplicamos dos for para el ingreso de datos que sería el nombre de los soldados usando arreglos.
- El código, la ejecución y el commit sería el siguiente:

Listing 7: Commit

```
$ git commit -m "Creamos el 3er ejercicio donde creamos 5 soldados y imprimimos sus
nombres usando arreglos"
```

Listing 8: Las líneas de código del Ejercicio03 serían

```
// Laboratorio Nro 1 - Ejercicio3
// Autor: Mamani Anahua Victor Narciso
// Colaboro:
// Tiempo:
import java.util.*;
public class Videojuego {
    public static void main(String args[]){
        //1. DECLARAMOS NUESTRO ARREGLO Y TAMBIEN LO INSTANCIAMOS
        Scanner sc = new Scanner(System.in);
```

```
String[] names = new String[5];
//2. LLENAMOS CADA ELEMENTO DEL ARREGLO
for(int i = 0; i < 5; i++){
    names[i] = sc.nextLine();
}
//3. MOSTRAMOS EL CONTENIDO DE CADA ELEMENTO DEL ARREGLO
for(int x = 0; x < 5 ; x++){
    System.out.println("El nombre del soldado numero " + (x + 1) + " es: " +
        names[x]);
}
}
```

Listing 9: La ejecución dada:

```
Ingrese su nombre:
JUAN
El nombre del soldado 1 es: JUAN
Ingrese su nombre:
BETO
El nombre del soldado 2 es: BETO
Ingrese su nombre:
TITO
El nombre del soldado 3 es: TITO
Ingrese su nombre:
HERNAN
El nombre del soldado 4 es: HERNAN
Ingrese su nombre:
HECTOR
El nombre del soldado 5 es: HECTOR
```

4.4. Ejercicio 04

- En el cuarto ejercicio en el commit especificamos y aplicamos dos for para el ingreso de datos que sería el nombre de los soldados y la vida de los soldados usando arreglos.
- El código, la ejecución y el commit sería el siguiente:

Listing 10: Commit

```
$ git commit -m "Creamos el 4to ejercicio donde creamos 5 soldados y imprimimos sus
nombres y su vida usando arreglos"
```

Listing 11: Las líneas de código del Ejercicio04 serían

```
// Laboratorio Nro 1 - Ejercicio4
// Autor: Mamani Anahua Victor Narciso
// Colaboro:
// Tiempo:
import java.util.*;
public class Videojuego {
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        //DECLARAMOS NUESTROS ARREGLOS
```

```
String[] name = new String[5];
int[] health = new int[5];
//CREAMOS UN CICLO PARA LA INFORMACION DE LOS 5 SOLDADOS
for(int i = 0; i < 5; i++){
    System.out.println("Ingresen sus nombres: ");
    name[i] = sc.next();
    System.out.println("Ingresen sus vidas:");
    health[i] = sc.nextInt();
}
//CREAMOS UN CICLO PARA DESPUES MOSTRARLOS
for(int x = 0; x < 5; x++){
    System.out.println("El nombre del soldado " + (x + 1) + " es: " + name[x]);
    System.out.println("La vida del soldado " + (x + 1) + " es: " + health[x]);
}
}
```

Listing 12: La ejecucion dada:

```
Ingresen sus nombres:
victor
Ingresen sus vidas:
45
Ingresen sus nombres:
hector
Ingresen sus vidas:
34
Ingresen sus nombres:
hernan
Ingresen sus vidas:
65
Ingresen sus nombres:
elsa
Ingresen sus vidas:
34
Ingresen sus nombres:
nar
Ingresen sus vidas:
23
El nombre del soldado 1 es: victor
La vida del soldado 1 es: 45
El nombre del soldado 2 es: hector
La vida del soldado 2 es: 34
El nombre del soldado 3 es: hernan
La vida del soldado 3 es: 65
El nombre del soldado 4 es: elsa
La vida del soldado 4 es: 34
El nombre del soldado 5 es: nar
La vida del soldado 5 es: 23
```

4.5. Ejercicio 05

- En el quinto ejercicio en el comit especificamos y aplicamos dos arrays de tipo string para despues usarlos en el codigo debido a su numero de soldados y aplicamos la estructura de control for para

sus nombres de los soldados y su impresion y tambien el uso de de un if para mostrar el resultado.

- El codigo , la ejecucion y el commit seria el siguiente:

Listing 13: Commit

```
$ git commit -m "En el 5toejercicio creamos 2 ejercitos y definimos un ganador o un  
empate dependiendo de su numero de soldados de cada ejercito"
```

Listing 14: Las lineas de codigo del Ejercicio04 serian

```
// Laboratorio Nro 1 - Ejercicio5  
// Autor: Mamani Anahua Victor Narciso  
// Colaboro:  
// Tiempo:  
import java.util.*;  
public class Videojuego{  
    public static void main(String args[]){  
        //DECLARAMOS Y INSTACIAMOS NUESTROS ARREGLOS  
        Random random = new Random();  
        String[] Army1 = new String[random.nextInt(5) + 1];  
        String[] Army2 = new String[random.nextInt(5) + 1];  
        //LLENAMOS CADA ARREGLO CON LO PEDIDO  
        for(int x = 0; x < Army1.length; x++){  
            Army1[x] = "Soldado" + x;  
        }  
        for(int y = 0; y < Army2.length; y++){  
            Army2[y] = "Soldado" + y;  
        }  
        //MOSTRAMOS LOS DATOS DE LOS SOLDADOS DE AMBOS EJERCITOS  
        System.out.println("LOS SOLDADOS DEL EJERCITO 1 SON: ");  
        for(int z = 0; z < Army1.length; z++){  
            System.out.println(Army1[z]);  
        }  
        System.out.println("LOS SOLDADOS DEL EJERCITO 2 SON: ");  
        for(int i = 0; i < Army2.length; i++){  
            System.out.println(Army2[i]);  
        }  
        //MOSTRAMOS EL GANADOR DE LA BATALLA  
        System.out.println("EL RESULTADO DE LA BATALLA ES :");  
        if(Army1.length > Army2.length){  
            System.out.println("EL EJERCITO 1 ES EL GANADOR");  
        }else{  
            if(Army1.length < Army2.length){  
                System.out.println("EL EJERCITO 2 ES EL GANADOR");  
            }else{  
                System.out.println("EMPATE");  
            }  
        }  
    }  
}
```

Listing 15: La ejecucion dada:

```
LOS SOLDADOS DEL EJERCITO 1 SON:
```

```
Soldado0
Soldado1
Soldado2
Soldado3
LOS SOLDADOS DEL EJERCITO 2 SON:
Soldado0
Soldado1
EL RESULTADO DE LA BATALLA ES :
EL EJERCITO 1 ES EL GANADOR
```

4.6. Estructura de laboratorio 01

- El contenido que se entrega en este laboratorio es el siguiente:

```
/Lab01
|----Latex
|    |----img
|    |    |----logo_abet.png
|    |    |----logo_episunsa.png
|    |    |----logo_unsa.jpg
|    |    +----pseudocodigo_insercion.png
|    |----InformeLab01.aux
|    |----InformeLab01.fdb_latexmk
|    |----InformeLab01.flx
|    |----InformeLab01.log
|    |----InformeLab01.out
|    |----InformeLab01.pdf
|    |----InformeLab01.synctex.gz
|    |----InformeLab01.tex
|    +----src
|    |    +----Videojuego01.java
+----Videojuego.java
```

5. Pregunta: antes de simular una batalla entre dos ejércitos, debemos considerar que cada ejército está compuesto por soldados. Dada su experiencia con videojuegos de estrategia, ¿qué datos de los soldados son importantes?

- Salud (Health o HP).
- Munición (Ammunition).
- Armadura (Armor).
- Nivel de experiencia (XP).
- Puntuación/Kill Count.

6. Rúbricas

6.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

6.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	1	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	0.5	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	2	
Total		20		12.5	

7. Referencias

- <https://drive.google.com/file/d/1KAJ3-N5uiKS0JaW0Tp0VycdIsxCYEXo3/view>