

Informe de Laboratorio 01

Tema: Arreglos estándar

Nota

Estudiante	Escuela	Asignatura
Christian Mestas Zegarra cmestasz@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación 2 Semestre: II Código: 1701213

Laboratorio	Tema	Duración
01	Arreglos estándar	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 13 Setiembre 2023	Al 20 Setiembre 2023

1. Tarea

- **Actividad 1:** 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.
- **Actividad 2:** 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.
- **Actividad 3:** Escribir un programa donde se creen 5 soldados considerando sólo su nombre. Ingresar sus datos y después mostrarlos. Restricción: aplicar arreglos estándar.
- **Actividad 4:** 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)
- **Actividad 5:** 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 Microsoft Windows 10 Pro 64 bits
- Visual Studio Code 1.82.2
- Java Development Kit 17.0.1
- Git 2.41.0.windows.1
- Windows PowerShell 5.1.19041.3031
- Cuenta en GitHub con el correo institucional.
- Arreglos estándar.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/cmestasz/fp2-23b.git>
- URL para el laboratorio 01 en el Repositorio GitHub.
- <https://github.com/cmestasz/fp2-23b/tree/main/fase01/lab01>

4. Actividades con el repositorio GitHub

4.1. Creando e inicializando repositorio GitHub

- Como es el primer laboratorio se creo el repositorio GitHub.
- Se realizaron los siguientes comandos en la computadora:

Creando directorio de trabajo

```
$ mkdir cmestasz  
$ cd cmestasz
```

Creando directorio para repositorio GitHub

```
$ mkdir fp2-23b  
$ cd fp2-23b  
$ mkdir fase01  
$ mkdir fase02  
$ mkdir fase03  
$ cd fase01  
$ mkdir lab01  
$ cd ..
```

Inicializando repositorio GitHub

```
$ git config --global user.name "Christian Mestas Zegarra"  
$ git config --global user.email cmestasz@unsa.edu.pe  
$ git init
```

4.2. Commits

Creando .gitignore

```
$ code .gitignore
```

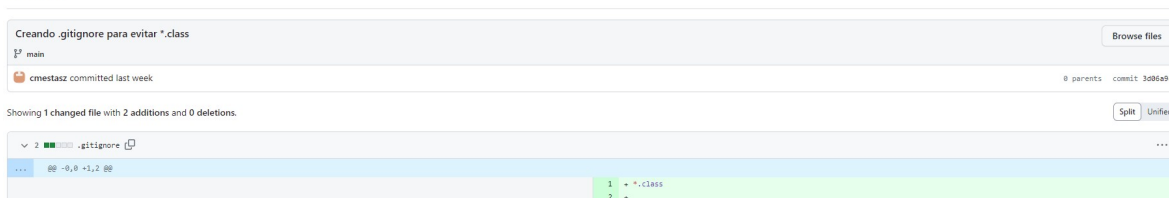
Primer Commit / .gitignore

```
$ git add .gitignore
$ git commit -m "Creando .gitignore para evitar *.class"
$ git branch -M main
$ git remote add origin https://github.com/cmestasz/fp2-23b.git
$ git push -u origin main
```

Los siguientes comandos son parte de la configuración del repositorio, por lo que solo son ejecutados durante el primer commit:

- **"git branch -M"** crea la rama main, en la que se trabajara posteriormente.
- **"git remote add origin"** configura la dirección del repositorio en línea.
- **"git push -u origin main"** configura la rama en la que se deben guardar los archivos.

Commit



Primer Commit.

Creando VideoJuego.java

```
$ cd fase01/lab01
$ code VideoJuego.java
```

Segundo Commit / VideoJuego.java

```
$ git add VideoJuego.java
$ git commit -m "Demo de VideoJuego.java donde saluda con un mensaje"
$ git push
```

Commit

```

Demo de VideoJuego.java donde saluda con un mensaje
main
cmestasz committed last week
1 parent 3d86a8c commit 3218f69

Showing 1 changed file with 6 additions and 0 deletions.
Split Unified

6 fse01/1a081/VideoJuego.java
...
1 public class VideoJuego {
2     public static void main(String[] args) {
3         System.out.println("Bienvenidos a mi VideoJuego");
4     }
5 }
6

```

Segundo Commit.

Creando Actividades.java

```
$ code Actividades.java
```

Tercer - Septimo Commit / Actividades.java

```

$ git add Actividades.java
$ git commit -m "Actividad 1"
$ code Actividades.java
$ git add Actividades.java
$ git commit -m "Actividad 2"
$ code Actividades.java
$ git add Actividades.java
$ git commit -m "Actividad 3"
$ code Actividades.java
$ git add Actividades.java
$ git commit -m "Actividad 4"
$ code Actividades.java
$ git add Actividades.java
$ git commit -m "Actividad 5"
$ git push

```

Commit

```

Actividad 1
main
cmestasz committed last week
1 parent 3218f69 commit 694f54f

Showing 1 changed file with 20 additions and 0 deletions.
Split Unified

20 fse01/1a081/Actividades.java
...
1 // Laboratorio Nro 1 - Actividad 1
2 // Autor: Christian Mestas
3
4 import java.util.Scanner;
5
6 public class Actividades {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         String nombreSoldado1 = sc.nextLine();
10        String nombreSoldado2 = sc.nextLine();
11        String nombreSoldado3 = sc.nextLine();
12        String nombreSoldado4 = sc.nextLine();
13        String nombreSoldado5 = sc.nextLine();
14        System.out.println("Soldado 1: " + nombreSoldado1);
15        System.out.println("Soldado 2: " + nombreSoldado2);
16        System.out.println("Soldado 3: " + nombreSoldado3);
17        System.out.println("Soldado 4: " + nombreSoldado4);
18        System.out.println("Soldado 5: " + nombreSoldado5);
19    }
20 }

```

Tercer Commit.

Commit

Actividad 2

main

cmestasz committed last week

Showing 1 changed file with 16 additions and 6 deletions.

Split Unify

@@ -1,4 +1,4 @@

```
1 - // Laboratorio Nro 1 - Actividad 1
2 // Autor: Christian Mestas
3
4 import java.util.Scanner;
```

```
1 + // Laboratorio Nro 1 - Actividad 1
2 // Autor: Christian Mestas
3
4 import java.util.Scanner;
```

```
7 public static void main(String[] args) {
8     Scanner sc = new Scanner(System.in);
9     String nombreSoldado1 = sc.nextLine();
10
11     String nombreSoldado2 = sc.nextLine();
12
13     String nombreSoldado3 = sc.nextLine();
14
15     String nombreSoldado4 = sc.nextLine();
16
17     String nombreSoldado5 = sc.nextLine();
18
19     System.out.println("Soldado 1: " + nombreSoldado1);
20     System.out.println("Soldado 2: " + nombreSoldado2);
21     System.out.println("Soldado 3: " + nombreSoldado3);
22     System.out.println("Soldado 4: " + nombreSoldado4);
23     System.out.println("Soldado 5: " + nombreSoldado5);
24 }
```

```
7 public static void main(String[] args) {
8     Scanner sc = new Scanner(System.in);
9     String nombreSoldado1 = sc.nextLine();
10     int videsSoldado1 = sc.nextInt();
11     sc.nextLine();
12     String nombreSoldado2 = sc.nextLine();
13     int videsSoldado2 = sc.nextInt();
14     sc.nextLine();
15     String nombreSoldado3 = sc.nextLine();
16     int videsSoldado3 = sc.nextInt();
17     sc.nextLine();
18     String nombreSoldado4 = sc.nextLine();
19     int videsSoldado4 = sc.nextInt();
20     sc.nextLine();
21     String nombreSoldado5 = sc.nextLine();
22     int videsSoldado5 = sc.nextInt();
23     sc.nextLine();
24     System.out.println("Soldado 1: " + nombreSoldado1 + ", Vides: " + videsSoldado1);
25     System.out.println("Soldado 2: " + nombreSoldado2 + ", Vides: " + videsSoldado2);
26     System.out.println("Soldado 3: " + nombreSoldado3 + ", Vides: " + videsSoldado3);
27     System.out.println("Soldado 4: " + nombreSoldado4 + ", Vides: " + videsSoldado4);
28     System.out.println("Soldado 5: " + nombreSoldado5 + ", Vides: " + videsSoldado5);
29 }
```

Cuarto Commit.

Commit

Actividad 3

main

cmestas committed last week

1 parent 49c4786 commit 5688a9

Showing 1 changed file with 8 additions and 21 deletions.

29 fse01/la001/Actividades.java

```

... @@ -1,30 +1,17 @@
1 - // Laboratorio Nro 1 - Actividad 2
2 - // Autor: Christian Nestas
3
4 - import java.util.Scanner;
5
6 - public class Actividades {
7 -     public static void main(String[] args) {
8 -         Scanner sc = new Scanner(System.in);
9 -
10 -         String nombreSoldado1 = sc.nextLine();
11 -         int vidSoldado1 = sc.nextInt();
12 -         sc.nextLine();
13 -         String nombreSoldado2 = sc.nextLine();
14 -         int vidSoldado2 = sc.nextInt();
15 -         sc.nextLine();
16 -         String nombreSoldado3 = sc.nextLine();
17 -         int vidSoldado3 = sc.nextInt();
18 -         sc.nextLine();
19 -         String nombreSoldado4 = sc.nextLine();
20 -         int vidSoldado4 = sc.nextInt();
21 -         sc.nextLine();
22 -         String nombreSoldado5 = sc.nextLine();
23 -         int vidSoldado5 = sc.nextInt();
24 -         sc.nextLine();
25 -         System.out.println("Soldado 1: " + nombreSoldado1 + ". Vida: " + vidSoldado1);
26 -         System.out.println("Soldado 2: " + nombreSoldado2 + ". Vida: " + vidSoldado2);
27 -         System.out.println("Soldado 3: " + nombreSoldado3 + ". Vida: " + vidSoldado3);
28 -         System.out.println("Soldado 4: " + nombreSoldado4 + ". Vida: " + vidSoldado4);
29 -         System.out.println("Soldado 5: " + nombreSoldado5 + ". Vida: " + vidSoldado5);
30     }
31 }
1 + // Laboratorio Nro 1 - Actividad 3
2 + // Autor: Christian Nestas
3
4 + import java.util.Scanner;
5
6 + public class Actividades {
7 +     public static void main(String[] args) {
8 +         Scanner sc = new Scanner(System.in);
9 +
10 +         String[] nombresSoldados = new String[5];
11 +         for (int i = 0; i < nombresSoldados.length; i++) {
12 +             nombresSoldados[i] = sc.nextLine();
13 +         }
14 +         for (int i = 0; i < nombresSoldados.length; i++) {
15 +             System.out.println("Soldado " + (i + 1) + ": " + nombresSoldados[i]);
16 +         }
17     }

```

Quinto Commit.

Commit

Actividad 4

main

cmetasz committed last week

1 parent 568Ba7f commit df3dc0c

Showing 1 changed file with 5 additions and 2 deletions.

▼

7

fase01/lab01/Actividades.java

... @@ -1,4 +1,4 @@

1 // Laboratorio Nro 1 - Actividad 4

2 // Autor: Christian Mestas

3

4 import java.util.Scanner;

5

6 @@ -7,11 +7,14 @@ public class Actividades {

7 public static void main(String[] args) {

8 Scanner sc = new Scanner(System.in);

9 String[] nombresSoldados = new String[5];

10

11 for (int i = 0; i < nombresSoldados.length; i++) {

nombresSoldados[i] = sc.nextLine();

12 }

13 for (int i = 0; i < nombresSoldados.length; i++) {

System.out.println("Soldado " + (i + 1) + ": " + nombresSoldados[i]);

15 }

16 }

17 }

18

19

20 }

1 // Laboratorio Nro 1 - Actividad 4

2 // Autor: Christian Mestas

3

4 import java.util.Scanner;

5

6 public static void main(String[] args) {

7 Scanner sc = new Scanner(System.in);

8 String[] nombresSoldados = new String[5];

9 int[] videsSoldados = new int[5];

10

11 for (int i = 0; i < nombresSoldados.length; i++) {

nombresSoldados[i] = sc.nextLine();

12

13 videsSoldados[i] = sc.nextInt();

14 sc.nextLine();

15 }

16 for (int i = 0; i < nombresSoldados.length; i++) {

17 System.out.println("Soldado " + (i + 1) + ": " + nombresSoldados[i] + ", Vidas: " + videsSoldados[i]);

18 }

19 }

20 }

Sexto Commit.

Commit

Actividad 5

main

cmrtest committed last week

1 parent d95d0a commit 5d3f2a

Showing 1 changed file with 30 additions and 11 deletions.

41

main/LAB01/Actividades.java

100 -1,20 +1,39

```
1 // Laboratorio 01 - Actividad 5
2 // Autor: Christian Nentes
3
4 - import java.util.Scanner;
5
6 public class Actividades {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         String[] nombresAlumnos = new String[10];
10         int[] vdsAlumnos = new int[10];
11         for (int i = 0; i < nombresAlumnos.length; i++) {
12             nombresAlumnos[i] = sc.nextLine();
13             vdsAlumnos[i] = sc.nextInt();
14             sc.nextLine();
15         }
16         for (int i = 0; i < nombresAlumnos.length; i++) {
17             System.out.println("Nombre" + (i + 1) + " = " + nombresAlumnos[i] + ", vds: " + vdsAlumnos[i]);
18         }
19     }
20 }
```

```
1 // Laboratorio 01 - Actividad 5
2 // Autor: Christian Nentes
3
4 + import java.util.Random;
5
6 public class Actividades {
7     public static void main(String[] args) {
8         String[] ejerCito = inicializarEjerCito();
9         String[] ejerCito2 = inicializarEjerCito();
10         mostrarEjerCito(ejerCito, 1);
11         mostrarEjerCito(ejerCito2, 2);
12         mostrarEjerCito(ejerCito, ejerCito2);
13     }
14
15     public static String[] inicializarEjerCito() {
16         Random random = new Random();
17         String[] ejerCito = new String[random.nextInt(5) + 1];
18         for (int i = 0; i < ejerCito.length; i++) {
19             ejerCito[i] = "Nombre" + i;
20         }
21         return ejerCito;
22     }
23
24     public static void mostrarEjerCito(String[] ejerCito, int numero) {
25         System.out.println("EjerCito " + numero + ":");
26         for (int i = 0; i < ejerCito.length; i++) {
27             System.out.println("Nombre" + (i + 1) + " = " + ejerCito[i]);
28         }
29     }
30
31     public static void mostrarEjerCito(String[] ejerCito1, String[] ejerCito2) {
32         if (ejerCito1.length > ejerCito2.length)
33             System.out.println("Solo el ejerCito 1!");
34         else if (ejerCito1.length < ejerCito2.length)
35             System.out.println("Solo el ejerCito 2!");
36         else
37             System.out.println("Hago un empate!");
38     }
39 }
```

Séptimo Commit.

Actualizando .gitignore



```
$ cd ../../..
$ code .gitignore
```

Octavo Commit / .gitignore



```
$ git add .gitignore
$ git commit -m "Actualizar .gitignore para que ignore los archivos generados por la
    compilacion latex"
$ git push
```

Commit

Actualizar .gitignore para que ignore los archivos generados por la c...
..ompilación latex [Browse files](#)

 main
 cmestas committed 16 hours ago 1 parent 04d013e commit c88d8b8

Showing 1 changed file with 2 additions and 0 deletions. [Split](#) [Unified](#)

▼ 2  .gitignore 

...	@@ -1,5 +1,7 @@	1	*.class
1	*.class	2	*.aux
2	*.aux	3	*.fda_latexmk
		4	*.fls
3	*.log	5	*.log
4	*.out	6	*.out
5	*.gz	7	*.gz
6			

Octavo Commit.

5. Código desarrollado

.gitignore

```
1 *.class
2 *.aux
3 *.log
4 *.out
5 *.gz
6 .DS_Store
```

Actividades.java (Actividad 1)

```
1 // Laboratorio Nro 1 - Actividad 1
2 // Autor: Christian Mestas
3
4 package fase01.lab01;
5
6 import java.util.Scanner;
7
8 public class Actividad1 {
9     public static void main(String[] args) {
10         Scanner sc = new Scanner(System.in);
11         String nombreSoldado1 = sc.nextLine();
12         String nombreSoldado2 = sc.nextLine();
13         String nombreSoldado3 = sc.nextLine();
14         String nombreSoldado4 = sc.nextLine();
15         String nombreSoldado5 = sc.nextLine();
16         System.out.println("Soldado 1: " + nombreSoldado1);
17         System.out.println("Soldado 2: " + nombreSoldado2);
18         System.out.println("Soldado 3: " + nombreSoldado3);
19         System.out.println("Soldado 4: " + nombreSoldado4);
20         System.out.println("Soldado 5: " + nombreSoldado5);
21     }
22 }
```

- Los 5 nombres son leídos línea por línea y guardados en 5 variables diferentes.
- Luego los datos son impresos línea por línea en el mismo orden en el que se ingresaron.

Actividades.java (Actividad 2)

```
1 // Laboratorio Nro 1 - Actividad 2
2 // Autor: Christian Mestas
3
4 package fase01.lab01;
5
6 import java.util.Scanner;
7
8 public class Actividad2 {
9     public static void main(String[] args) {
10         Scanner sc = new Scanner(System.in);
11         String nombreSoldado1 = sc.nextLine();
12         int vidaSoldado1 = sc.nextInt();
13         sc.nextLine();
14         String nombreSoldado2 = sc.nextLine();
15         int vidaSoldado2 = sc.nextInt();
16         sc.nextLine();
17         String nombreSoldado3 = sc.nextLine();
18         int vidaSoldado3 = sc.nextInt();
19         sc.nextLine();
20         String nombreSoldado4 = sc.nextLine();
21         int vidaSoldado4 = sc.nextInt();
22         sc.nextLine();
23         String nombreSoldado5 = sc.nextLine();
24         int vidaSoldado5 = sc.nextInt();
25         sc.nextLine();
26         System.out.println("Soldado 1: " + nombreSoldado1 + ". Vida: " + vidaSoldado1);
27         System.out.println("Soldado 2: " + nombreSoldado2 + ". Vida: " + vidaSoldado2);
28         System.out.println("Soldado 3: " + nombreSoldado3 + ". Vida: " + vidaSoldado3);
29         System.out.println("Soldado 4: " + nombreSoldado4 + ". Vida: " + vidaSoldado4);
30         System.out.println("Soldado 5: " + nombreSoldado5 + ". Vida: " + vidaSoldado5);
31     }
32 }
```

- Los 5 nombres y sus respectivas vidas son leídas línea por línea y guardadas en 5 variables diferentes por atributo.
- Luego los datos son impresos línea por línea en el mismo orden en el que se ingresaron.

Actividades.java (Actividad 3)

```
1 // Laboratorio Nro 1 - Actividad 3
2 // Autor: Christian Mestas
3
4 package fase01.lab01;
5
6 import java.util.Scanner;
7
8 public class Actividad3 {
9     public static void main(String[] args) {
10         Scanner sc = new Scanner(System.in);
11         String[] nombresSoldados = new String[5];
12         for (int i = 0; i < nombresSoldados.length; i++) {
13             nombresSoldados[i] = sc.nextLine();
14         }
15         for (int i = 0; i < nombresSoldados.length; i++) {
16             System.out.println("Soldado " + (i + 1) + ": " + nombresSoldados[i]);
17         }
18     }
19 }
```

- Los 5 nombres son leídos iterativamente y guardados en un arreglo.
- Luego los datos son impresos iterativamente en el mismo orden en el que se ingresaron.

Actividades.java (Actividad 4)

```
1 // Laboratorio Nro 1 - Actividad 4
2 // Autor: Christian Mestas
3
4 package fase01.lab01;
5
6 import java.util.Scanner;
7
8 public class Actividad4 {
9     public static void main(String[] args) {
10         Scanner sc = new Scanner(System.in);
11         String[] nombresSoldados = new String[5];
12         int[] vidasSoldados = new int[5];
13         for (int i = 0; i < nombresSoldados.length; i++) {
14             nombresSoldados[i] = sc.nextLine();
15             vidasSoldados[i] = sc.nextInt();
16             sc.nextLine();
17         }
18         for (int i = 0; i < nombresSoldados.length; i++) {
19             System.out.println("Soldado " + (i + 1) + ": " + nombresSoldados[i] + ". Vida: " +
20                               vidasSoldados[i]);
21         }
22     }
23 }
```

- Los 5 nombres y sus respectivas vidas son leídas iterativamente y guardadas en 2 arreglos.
- Luego los datos son impresos iterativamente en el mismo orden en el que se ingresaron.

Actividades.java (Actividad 5)

```
1 // Laboratorio Nro 1 - Actividad 5
2 // Autor: Christian Mestas
3
4 package fase01.lab01;
5
6 import java.util.Random;
7
8 public class Actividad5 {
9     public static void main(String[] args) {
10         String[] ejercito1 = inicializarEjercito();
11         String[] ejercito2 = inicializarEjercito();
12         mostrarEjercito(ejercito1, 1);
13         mostrarEjercito(ejercito2, 2);
14         mostrarGanador(ejercito1, ejercito2);
15     }
16
17     public static String[] inicializarEjercito() {
18         Random random = new Random();
19         String[] ejercito = new String[random.nextInt(5) + 1];
20         for (int i = 0; i < ejercito.length; i++) {
21             ejercito[i] = "Soldado" + i;
22         }
23         return ejercito;
24     }
25
26     public static void mostrarEjercito(String[] ejercito, int numero) {
27         System.out.println("Ejercito " + numero + ":");
28         for (int i = 0; i < ejercito.length; i++) {
29             System.out.println("Soldado " + (i + 1) + ": " + ejercito[i]);
30         }
31     }
32
33     public static void mostrarGanador(String[] ejercito1, String[] ejercito2) {
34         if (ejercito1.length > ejercito2.length)
35             System.out.println("Gana el ejercito 1!");
36         else if (ejercito1.length < ejercito2.length)
37             System.out.println("Gana el ejercito 2!");
38         else
39             System.out.println("Hubo un empate!");
40     }
41 }
```

- Se crean 2 ejércitos con el método `inicializarEjercito()`, que crea un ejército de tamaño al azar entre 1 y 5.
- Se muestran los ejércitos con el método `mostrarEjercito()`, que imprime una lista de los soldados del ejército.
- Se muestra el ganador con el método `mostrarGanador()`, que compara los tamaños de los ejércitos y declara al ganador como el ejército mas grande.

6. Ejecución del código

```
Windows PowerShell
PS F:\Documents\UNSA\A1S2\FP2 lab\fp2-23b\fase01\lab01\src> java Actividad1
Perez Jose
Luis Luis
Juan Oz
Marco Pedro
Pedro Marco
Soldado 1: Perez Jose
Soldado 2: Luis Luis
Soldado 3: Juan Oz
Soldado 4: Marco Pedro
Soldado 5: Pedro Marco
PS F:\Documents\UNSA\A1S2\FP2 lab\fp2-23b\fase01\lab01\src>
```

Actividad 1

```
Windows PowerShell
PS F:\Documents\UNSA\A1S2\FP2 lab\fp2-23b\fase01\lab01\src> java Actividad2
Perez Jose
50
Luis Luis
45
Juan Oz
55
Marco Pedro
125
Pedro Marco
30
Soldado 1: Perez Jose. Vida: 50
Soldado 2: Luis Luis. Vida: 45
Soldado 3: Juan Oz. Vida: 55
Soldado 4: Marco Pedro. Vida: 125
Soldado 5: Pedro Marco. Vida: 30
PS F:\Documents\UNSA\A1S2\FP2 lab\fp2-23b\fase01\lab01\src>
```

Actividad 2

```
Windows PowerShell
PS F:\Documents\UNSA\A1S2\FP2 lab\fp2-23b\fase01\lab01\src> java Actividad3
Perez Jose
Luis Luis
Juan Oz
Marco Pedro
Pedro Marco
Soldado 1: Perez Jose
Soldado 2: Luis Luis
Soldado 3: Juan Oz
Soldado 4: Marco Pedro
Soldado 5: Pedro Marco
PS F:\Documents\UNSA\A1S2\FP2 lab\fp2-23b\fase01\lab01\src>
```

Actividad 3

```
Windows PowerShell
PS F:\Documents\UNSA\A1S2\FP2 lab\fp2-23b\fase01\lab01\src> java Actividad4
Perez Jose
50
Luis Luis
45
Juan Oz
55
Marco Pedro
125
Pedro Marco
30
Soldado 1: Perez Jose. Vida: 50
Soldado 2: Luis Luis. Vida: 45
Soldado 3: Juan Oz. Vida: 55
Soldado 4: Marco Pedro. Vida: 125
Soldado 5: Pedro Marco. Vida: 30
PS F:\Documents\UNSA\A1S2\FP2 lab\fp2-23b\fase01\lab01\src>
```

Actividad 4

```
Windows PowerShell
PS F:\Documents\UNSA\AIS2\FP2 lab\fp2-23b\fase01\lab01\src> java Actividad5
Ejercito 1:
Soldado 1: Soldado0
Soldado 2: Soldado1
Soldado 3: Soldado2
Soldado 4: Soldado3
Ejercito 2:
Soldado 1: Soldado0
Gana el ejercito 1!
PS F:\Documents\UNSA\AIS2\FP2 lab\fp2-23b\fase01\lab01\src>
```

Actividad 5

7. Estructura de laboratorio 01

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

```
lab01/  
|--- VideoJuego.java  
|--- Actividades.java  
|--- Informe.tex  
|--- Informe.pdf  
|--- img  
|   |--- logo_abet.png  
|   |--- logo_episunsa.png  
|   |--- logo_unsa.jpg  
|   |--- commit01.jpg  
|   |--- commit02.jpg  
|   |--- commit03.jpg  
|   |--- commit04.jpg  
|   |--- commit05.jpg  
|   |--- commit06.jpg  
|   |--- commit07.jpg  
|   |--- commit08.jpg  
|   |--- ejec01.jpg  
|   |--- ejec02.jpg  
|   |--- ejec03.jpg  
|   |--- ejec04.jpg  
|   |--- ejec05.jpg  
|--- src  
|   |--- .gitignore  
|   |--- Actividad1.java  
|   |--- Actividad2.java  
|   |--- Actividad3.java  
|   |--- Actividad4.java  
|   |--- Actividad5.java
```

8. Rúbricas

8.1. Entregable Informe

Tipo de Informe

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

8.2. Rúbrica para el contenido del Informe y demostración

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

Niveles de desempeño

Puntos	Nivel			
	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

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	4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1.5	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2		0	
6. Fechas	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
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	3	
Total		20		16.5	

9. Referencias

- Aedo, M. y Castro, E. (2021). FUNDAMENTOS DE PROGRAMACIÓN 2 - Tópicos de Programación Orientada a Objetos. Editorial UNSA.