
	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	<i>Fundamentos de la programación 2</i>				
TÍTULO DE LA PRÁCTICA:	<i>Laboratorio 3</i>				
NÚMERO DE PRÁCTICA:	<i>3</i>	AÑO LECTIVO:	<i>1</i>	NRO. SEMESTRE:	<i>2</i>
FECHA DE PRESENTACIÓN	<i>06/10/2024</i>	HORA DE PRESENTACIÓN	<i>01/25/00</i>		
INTEGRANTE (s) Karla Miluska Bedregal Coaguila				NOTA (0-20)	
DOCENTE(s): <i>Lino Jose Pinto Oppe</i>					

RESULTADOS Y PRUEBAS
<p>I. EJERCICIOS RESUELTOS:</p> <p>1. Analice, complete y pruebe el Código de la clase Nave</p>

```
1  /* Autor: Karla Bedregal Coaguila
2   Analice, complete y pruebe el Código de la clase Nave */
3  package laboratorio03_karlabedregal;
4  import java.util.*;
5  public class Nave {
6      private String nombre;
7      private int fila;
8      private String columna;
9      private boolean estado;
10     private int puntos;
11     // Metodos mutadores
12     public void setNombre( String n){
13         nombre = n;
14     }
15     public void setFila(int f){
16         fila = f;
17     }
18     public void setColumna(String c){
19         columna = c;
20     }
21     public void setEstado(boolean e){
22         estado = e;
23     }
24     public void setPuntos(int p){
25         puntos = p;
26     }
27     // Metodos accesoros
28     public String getNombre(){
29         return nombre;
30     }
31     public int getFila(){
32         return fila;
33     }
34     public String getColumna(){
35         return columna;
36     }
37     public boolean getEstado(){
38         return estado;
39     }
40     public int getPuntos(){
41         return puntos;
42     }
43     // Completar con otros métodos necesarios
44     public String toString() {
45         return "Nave: \nNombre->" + nombre + ", Fila->" + fila + ", Columna->"
46             + columna + ", Estado->" + estado + ", Puntos->" + puntos;
47     }
48 }
```

2. Analice, complete y pruebe el Código de la clase DemoBatalla

```
1  /* Autor: Karla Bedregal Coaguila
2  |  Analice, complete y pruebe el Código de la clase DemoBatalla */
3  package laboratorio03_karlabedregal;
4  import java.util.*;
5  public class DemoBatalla {
6      public static void main(String [] args){
7          Nave [] misNaves = new Nave[10];
8          Scanner sc = new Scanner(System.in);
9          String nomb, col;
10         int fil, punt;
11         boolean est;
12
13         for (int i = 0; i < misNaves.length; i++) {
14             System.out.println("Nave " + (i+1));
15             System.out.print("Nombre: ");
16             nomb = sc.next();
17             System.out.println("Fila ");
18             fil = sc.nextInt();
19             System.out.print("Columna: ");
20             col = sc.next();
21             System.out.print("Estado: ");
22             est = sc.nextBoolean();
23             System.out.print("Puntos: ");
24             punt = sc.nextInt();
25
26             misNaves[i] = new Nave(); //Se crea un objeto Nave
27
28             misNaves[i].setNombre(nomb);
29             misNaves[i].setFila(fil);
30             misNaves[i].setColumna(col);
31             misNaves[i].setEstado(est);
32             misNaves[i].setPuntos(punt);
33         }
34
35         System.out.println("\nNaves creadas:");
36         mostrarNaves(misNaves);
37         mostrarPorNombre(misNaves);
38         mostrarPorPuntos(misNaves);
39         System.out.println("\nNave con mayor número de puntos: " + mostrarMayorPuntos(misNaves));
40         Nave[] flotaDesordenada = desordenarNaves(misNaves);
41         System.out.println("Arreglo desordenado:");
42         mostrarNaves(flotaDesordenada);
43     }
44     //Método para mostrar todas las naves
45     public static void mostrarNaves(Nave [] flota){
46         System.out.println("Naves:");
47         //Usamos un for-each para imprimir
48         for (Nave c : flota) {
49             System.out.println(c);
50         }
51     }
```

```

31 //Método para mostrar todas las naves de un nombre que se pide por teclado
32 public static void mostrarPorNombre(Nave [] flota){
33     Scanner sc = new Scanner (System.in);
34     System.out.println("Ingrese el nombre: ");
35     String nombreIngresado = sc.next();
36     //Un for para mostrar las filas, columnas, estado y puntos del nombre
37     for (int i = 0; i < flota.length; i++) {
38         if (nombreIngresado.equalsIgnoreCase(flota[i].getNombre())) {
39             System.out.println(flota[i].getFila());
40             System.out.println(flota[i].getColumna());
41             System.out.println(flota[i].getEstado());
42             System.out.println(flota[i].getPuntos());
43         }
44     }
45 }
46 //Método para mostrar todas las naves con un número de puntos inferior o igual
47 //al número de puntos que se pide por teclado
48 public static void mostrarPorPuntos(Nave [] flota){
49     Scanner sc = new Scanner (System.in);
50     System.out.println("Ingrese el número de puntos: ");
51     int numPuntos = sc.nextInt();
52     for (int i = 0; i < flota.length; i++) {
53         if (flota[i].getPuntos() <= numPuntos) {
54             System.out.println(flota[i].getNombre());
55         }
56     }
57 }
58 //Método que devuelve la Nave con mayor número de Puntos
59 public static Nave mostrarMayorPuntos(Nave [] flota){
60     Scanner sc = new Scanner (System.in);
61     Nave mayor = flota[0];
62     for (int i = 1; i < flota.length; i++) {
63         if (flota[i].getPuntos() > mayor.getPuntos()) {
64             mayor = flota[i]; // Si encontramos una nave con más puntos, actualizamos "mayor"
65         }
66     }
67     return mayor; // Devolvemos la nave con más puntos
68 }
69 //Crear un método que devuelva un nuevo arreglo de objetos con todos los objetos previamente
70 // ingresados pero aleatoriamente desordenados
71 public static Nave[] desordenarNaves(Nave[] flota) {
72     Nave[] flotaDesordenada = Arrays.copyOf(flota, flota.length); // Copiamos el arreglo original
73     Random rand = new Random();
74     for (int i = flotaDesordenada.length - 1; i > 0; i--) {
75         int j = rand.nextInt(i + 1); // Generamos un random
76         // Intercambiamos flotaDesordenada[i] con flotaDesordenada[j]
77         Nave des = flotaDesordenada[i];
78         flotaDesordenada[i] = flotaDesordenada[j];
79         flotaDesordenada[j] = des;
80     }
81     return flotaDesordenada; // Arreglo desordenado
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }

```

3. Solucionar la Actividad 4 de la Práctica 1 pero usando arreglo de objetos

```
1  /* Autor: Karla Bedregal Coaguila
2  Solucionar la Actividad 4 de la Práctica 1 pero usando arreglo de objetos */
3  package laboratorio03_karlabedregal;
4  public class Soldado {
5      private String nombre;
6      private int nivelVida;
7      // Constructor
8      public Soldado(String nombre, int nivelVida) {
9          this.nombre = nombre;
10         this.nivelVida = nivelVida;
11     }
12     // Métodos accesorios
13     public String getNombre() {
14         return nombre;
15     }
16     public int getNivelVida() {
17         return nivelVida;
18     }
19     // Método para mostrar información del soldado
20     public String toString() {
21         return "Nombre: " + nombre + ", Nivel de vida: " + nivelVida;
22     }
23 }
```

```
1  /* Autor: Karla Bedregal Coaguila
2  Solucionar la Actividad 4 de la Práctica 1 pero usando arreglo de objetos */
3  package laboratorio03_karlabedregal;
4  import java.util.Scanner;
5  public class Ejercicio4 {
6      public static void main(String[] args) {
7          Scanner sc = new Scanner(System.in);
8
9          // Arreglo de soldados
10         Soldado[] soldados = new Soldado[5];
11
12         // Ingresar nombre y nivel de vida
13         for (int i = 0; i < soldados.length; i++) {
14             System.out.println("Ingrese el nombre del soldado " + (i + 1) + ": ");
15             String nombre = sc.nextLine();
16
17             System.out.println("Ingrese el nivel de vida del soldado " + (i + 1) + ": ");
18             int nivelVida = sc.nextInt();
19             sc.nextLine(); // Limpiar el buffer
20
21             // Crear el objeto Soldado y almacenarlo en el arreglo
22             soldados[i] = new Soldado(nombre, nivelVida);
23         }
24
25         // Imprimir
26         System.out.println("\nInformación de los soldados:");
27         for (Soldado soldado : soldados) {
28             System.out.println(soldado); // Llama al método toString
29         }
30     }
31 }
```

4. Solucionar la Actividad 5 de la Práctica 1 pero usando arreglos de objetos

```
1  /* Autor: Karla Bedregal Coaguila
2  Solucionar la Actividad 5 de la Práctica 1 pero usando arreglo de objetos */
3  package laboratorio03_karlabedregal;
4  public class Soldado1 {
5      private String nombre;
6      // Constructor
7      public Soldado1(String nombre) {
8          this.nombre = nombre;
9      }
10     // Método accesorios
11     public String getNombre() {
12         return nombre;
13     }
14     public String toString() {
15         return nombre;
16     }
17 }
```





```
1  /* Autor: Karla Bedregal Coaguila
2  Solucionar la Actividad 5 de la Práctica 1 pero usando arreglo de objetos */
3  package laboratorio03_karlabedregal;
4  import java.util.Random;
5  public class Ejercicio5 {
6      public static void main(String[] args) {
7          Random rand = new Random();
8          // Número de soldados random para los equipos
9          int numeroSoldadosA = rand.nextInt(5) + 1;
10         int numeroSoldadosB = rand.nextInt(5) + 1;
11         // Arreglos para los soldados
12         Soldado1[] soldadosA = new Soldado1[numeroSoldadosA];
13         Soldado1[] soldadosB = new Soldado1[numeroSoldadosB];
14         // Asignar nombres a los soldados del equipo A
15         for (int i = 0; i < numeroSoldadosA; i++) {
16             soldadosA[i] = new Soldado1("SoldadoA" + (i + 1));
17         }
18         // Asignar nombres a los soldados del equipo B
19         for (int j = 0; j < numeroSoldadosB; j++) {
20             soldadosB[j] = new Soldado1("SoldadoB" + (j + 1));
21         }
22         // Métodos
23         System.out.println("\nSoldados del ejército A: " + numeroSoldadosA);
24         mostrarEjercito(soldadosA);
25         System.out.println("\nSoldados del ejército B: " + numeroSoldadosB);
26         mostrarEjercito(soldadosB);
27         ejercitoGanador(numeroSoldadosA, numeroSoldadosB);
28     }
29     public static void mostrarEjercito(Soldado1[] array) {
30         for (Soldado1 soldado : array) {
31             System.out.println("Nombre de los soldados: " + soldado.getNombre());
32         }
33     }
34     public static void ejercitoGanador(int a, int b) {
35         if (a == b) {
36             System.out.println("\nEMPATE - No hay ganador");
37         } else if (a < b) {
38             System.out.println("\nGanador - EJERCITO B con " + b + " soldados");
39         } else {
40             System.out.println("\nGanador - EJERCITO A con " + a + " soldados");
41         }
42     }
43 }
```

II. PRUEBAS

Ejecución 1 (Nave y DemoBatalla):

```
run:
Nave 1
Nombre: grul
Fila
4
Columna: 2
Estado: true
Puntos: 54
Nave 2
Nombre: frill
Fila
3
Columna: 4
Estado: false
Puntos: 65
Nave 3
Nombre: drill
Fila
4
Columna: 1
Estado: false
Puntos: 25
Nave 4
Nombre: olgita
Fila
2
Columna: 6
Estado: true
Puntos: 13
Nave 5
Nombre: freya
Fila
3
Columna: 7
Estado: true
Puntos: 14
```

```
Nave 6
Nombre: nuevo
Fila
3
Columna: 8
Estado: true
Puntos: 29
Nave 7
Nombre: gruñom
Fila
7
Columna: 6
Estado: true
Puntos: 29
Nave 8
Nombre: fresa
Fila
2
Columna: 7
Estado: false
Puntos: 19
Nave 9
Nombre: clandi
Fila
2
Columna: 6
Estado: false
Puntos: 64
Nave 10
Nombre: mana
Fila
9
Columna: 4
Estado: false
Puntos: 84
```


	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 9</p>

Naves creadas:

Naves:

Nave:

Nombre->grul, Fila->4, Columna->2, Estado->>true, Puntos->54

Nave:

Nombre->frill, Fila->3, Columna->4, Estado->>false, Puntos->65

Nave:

Nombre->drill, Fila->4, Columna->1, Estado->>false, Puntos->25

Nave:

Nombre->olgita, Fila->2, Columna->6, Estado->>true, Puntos->13

Nave:

Nombre->freya, Fila->3, Columna->7, Estado->>true, Puntos->14

Nave:

Nombre->nuevo, Fila->3, Columna->8, Estado->>true, Puntos->29

Nave:

Nombre->gruom, Fila->7, Columna->6, Estado->>true, Puntos->29

Nave:

Nombre->fresa, Fila->2, Columna->7, Estado->>false, Puntos->19

Nave:

Nombre->clandi, Fila->2, Columna->6, Estado->>false, Puntos->64

Nave:

Nombre->mana, Fila->9, Columna->4, Estado->>false, Puntos->84

Ingrese el nombre:

freya

3

7

true

14

Ingrese el número de puntos:

29

drill



olgita

freya

nuevo

gruom



fresa

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p style="text-align: center;">Código: GUIA-PRLE-001</p>	<p style="text-align: right;">Página: 10</p>

```

Nave con mayor número de puntos: Nave:
Nombre->mana, Fila->9, Columna->4, Estado->>false, Puntos->84
Arreglo desordenado:
Naves:
Nave:
Nombre->drill, Fila->4, Columna->1, Estado->>false, Puntos->25
Nave:
Nombre->freya, Fila->3, Columna->7, Estado->>true, Puntos->14
Nave:
Nombre->grul, Fila->4, Columna->2, Estado->>true, Puntos->54
Nave:
Nombre->gruom, Fila->7, Columna->6, Estado->>true, Puntos->29
Nave:
Nombre->olgita, Fila->2, Columna->6, Estado->>true, Puntos->13
Nave:
Nombre->frill, Fila->3, Columna->4, Estado->>false, Puntos->65
Nave:
Nombre->mana, Fila->9, Columna->4, Estado->>false, Puntos->84
Nave:
Nombre->fresa, Fila->2, Columna->7, Estado->>false, Puntos->19
Nave:
Nombre->nuevo, Fila->3, Columna->8, Estado->>true, Puntos->29
Nave:
Nombre->clandi, Fila->2, Columna->6, Estado->>false, Puntos->64
BUILD SUCCESSFUL (total time: 2 minutes 42 seconds)

```

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 11</p>

Ejecución 2 (Ejercicio 4 con objetos):

```
run:
Ingrese el nombre del soldado 1:
Juan
Ingrese el nivel de vida del soldado 1:
4
Ingrese el nombre del soldado 2:
Lauro
Ingrese el nivel de vida del soldado 2:
9
Ingrese el nombre del soldado 3:
Jonas
Ingrese el nivel de vida del soldado 3:
3
Ingrese el nombre del soldado 4:
Rodri
Ingrese el nivel de vida del soldado 4:
6
Ingrese el nombre del soldado 5:
Esteban
Ingrese el nivel de vida del soldado 5:
7

Información de los soldados:
Nombre: Juan, Nivel de vida: 4
Nombre: Lauro, Nivel de vida: 9
Nombre: Jonas, Nivel de vida: 3
Nombre: Rodri, Nivel de vida: 6
Nombre: Esteban, Nivel de vida: 7
BUILD SUCCESSFUL (total time: 43 seconds)
```



Ejecución 3 (Ejercicio 5 con objetos):

```
run:

Soldados del ejército A: 5
Nombre de los soldados: SoldadoA1
Nombre de los soldados: SoldadoA2
Nombre de los soldados: SoldadoA3
Nombre de los soldados: SoldadoA4
Nombre de los soldados: SoldadoA5

Soldados del ejército B: 2
Nombre de los soldados: SoldadoB1
Nombre de los soldados: SoldadoB2

Ganador - EJERCITO A con 5 soldados
BUILD SUCCESSFUL (total time: 0 seconds)
```

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 12</p>

¿Con que valores comprobaste que tu práctica estuviera correcta?

Con Strings, enteros y booleanos. Luego de colocar los nombres de soldados, naves, niveles de vida, etc, durante la ejecución se verifica si el programa esta funcionando bien. Si los elementos de salida son los esperados, entonces ya no hay necesidad de cambiar el código.

¿Qué resultado esperabas obtener para cada valor de entrada?

Esperaba que cada método funcionara, que imprimieran, que compararan elementos como para hallar el mayor en un arreglo de objetos, que desordenaran creando un arreglo copia, etc.

¿Qué valor o comportamiento obtuviste para cada valor de entrada?

Algunos se me hicieron complicados, pero luego de revisar libros y páginas de internet pude encontrar la manera de que funcionaran mejor.

III. CUESTIONARIO:

Conectar con github:

Primero iniciamos git (git init) en la ruta donde queremos subir los archivos. A continuación, verificamos con git status que no hay commits. Luego, con git add . agregamos todos los archivos que tenemos en la carpeta.

Configuramos en powershell el nombre de usuario, y correo electrónico usando git user.name y git user.email. Luego, hacemos el commit con git commit -m "y un mensaje más".

Finalmente, con git remote add origin (y añadimos el link de nuestro repositorio).

```
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git init
Initialized empty Git repository in C:/Users/LENOVO/Documents/NetBeansProjects/Lab01_fp2-KarlaBedregal/Laboratorio03_KarlaBedregal/src/.git/
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> ls

Directorio: C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src

Mode                LastWriteTime         Length Name
----                -
d-----           5/10/2024    23:53             laboratorio03_karlabedregal

PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    laboratorio03_karlabedregal/



nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git add .
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   laboratorio03_karlabedregal/DemoBatalla.java

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   laboratorio03_karlabedregal/DemoBatalla.java
    new file:   laboratorio03_karlabedregal/Ejercicio4.java
    new file:   laboratorio03_karlabedregal/Ejercicio5.java
    new file:   laboratorio03_karlabedregal/Have.java
    new file:   laboratorio03_karlabedregal/Soldado.java
    new file:   laboratorio03_karlabedregal/Soldado1.java

PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git user.name "KarlaBedregal"
git: 'user.name' is not a git command. See 'git --help'.
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git config user.name "KarlaBedregal"
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git config user.email "kbedregalc@unsa.edu.pe"
```

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 14</p>

```

user.name=KarlaBedregal
user.email=kbedregalc@unsa.edu.pe
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git commit -m "Subiendo archivos de lab 03"
[master (root-commit) ed0fa20] Subiendo archivos de lab 03
6 files changed, 269 insertions(+)
create mode 100644 laboratorio03_karlabedregal/DemoBatalla.java
create mode 100644 laboratorio03_karlabedregal/Ejercicio4.java
create mode 100644 laboratorio03_karlabedregal/Ejercicio5.java
create mode 100644 laboratorio03_karlabedregal/Nave.java
create mode 100644 laboratorio03_karlabedregal/Soldado.java
create mode 100644 laboratorio03_karlabedregal/Soldado1.java
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git remote add origin https://github.com/KarlaBedregal/fundamentos2-lab03.git
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/KarlaBedregal/fundamentos2-lab03.git'
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src> git push -u origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 3.58 KiB | 610.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/KarlaBedregal/fundamentos2-lab03.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\Laboratorio03_KarlaBedregal\src>

```

CONCLUSIONES

Al finalizar este laboratorio, hemos explorado los arreglos de objetos de este programa, haciendo uso de métodos, clases y atributos de arreglos. Al ser un código proporcionado parcialmente, solo faltaron algunos ajustes para completar y verificar si funcionaba correctamente.



Además, hemos cambiado los ejercicios del laboratorio 1, a arreglo de objetos para aprender mejor sobre ellos, por eso, después de finalizar, concluimos que este laboratorio fue muy útil pues puso en práctica de todo lo aprendido hasta hoy en el curso.

METODOLOGÍA DE TRABAJO

El procedimiento usado para este laboratorio es el siguiente:

- Primero analizamos el enunciado/actividad.
- Verificamos e importamos el paquete que vamos a necesitar.
- Luego definimos los datos que nos presenta y/o creamos las variables.
- Aplicamos la lógica de programación al hacer los programas.
- Creamos y/o actualizamos los métodos.
- Nos aseguramos que no muestre error.
- Al finalizar, ejecutamos y revisamos si funciona bien.

Contenido y demostración		Puntos	Checklis t	Estudiant e	Profeso r
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	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	1	
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		17	

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 16</p>

REFERENCIAS Y BIBLIOGRAFÍA
<p>https://github.com/LINOPINTO2023/FundProg2/tree/main/IEEE</p>