


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

## INFORME DE LABORATORIO

### (formato estudiante)

INFORMACIÓN BÁSICA					
<b>ASIGNATURA:</b>	<i>Fundamentos de la programación 2</i>				
<b>TÍTULO DE LA PRÁCTICA:</b>	<i>Laboratorio 4</i>				
<b>NÚMERO DE PRÁCTICA:</b>	<i>4</i>	<b>AÑO LECTIVO:</b>	<i>1</i>	<b>NRO. SEMESTRE:</b>	<i>2</i>
<b>FECHA DE PRESENTACIÓN</b>	<i>13/10/2024</i>	<b>HORA DE PRESENTACIÓN</b>	<i>22/15/00</i>		
<b>INTEGRANTE (s)</b> Karla Miluska Bedregal Coaguila				<b>NOTA (0-20)</b>	
<b>DOCENTE(s):</b> <i>Lino Jose Pinto Oppe</i>					

RESULTADOS Y PRUEBAS	
<b>I. EJERCICIOS RESUELTOS:</b>	
1. Clase Nave	
<pre> 1  /* Autor: Karla Bedregal Coaguila 2  Laboratorio 04 - Ejercicio 01 3  Analice, complete y pruebe el Código de la clase Nave */ 4  package laboratorio04_karlabledregal; 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     } </pre>	

	<p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<p style="text-align: center;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 2</p>

```

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. Clase DemoBatalla

```

1  /* Autor: Karla Bedregal Coaguila
2   Laboratorio 04 - Ejercicio 01
3   Usted podrá reutilizar las dos clases Nave.java y DemoBatalla.java. creadas en
4   Laboratorio 3. Completar el código de la clase DemoBatalla*/
5  package laboratorio04_karlabedregal;
6  import java.util.*;
7  public class DemoBatalla {
8      public static void main(String [] args){
9          Nave [] misNaves = new Nave[8];
10         Scanner sc = new Scanner(System.in);
11         String nomb, col;
12         int fil, punt;
13         boolean est;
14
15         for (int i = 0; i < misNaves.length; i++) {
16             System.out.println("Nave " + (i+1));
17             System.out.print("Nombre: ");
18             nomb = sc.next();
19             System.out.print("Fila: ");
20             fil = sc.nextInt();
21             System.out.print("Columna: ");
22             col = sc.next();
23             System.out.print("Estado: ");
24             est = sc.nextBoolean();
25             System.out.print("Puntos: ");
26             punt = sc.nextInt();

```

```
27      misNaves[i] = new Nave(); //Se crea un objeto Nave
28
29
30      misNaves[i].setNombre(nomb);
31      misNaves[i].setFila(fil);
32      misNaves[i].setColumna(col);
33      misNaves[i].setEstado(est);
34      misNaves[i].setPuntos(punt);
35  }
36
37  System.out.println("\nNaves creadas:");
38  mostrarNaves(misNaves);
39  mostrarPorNombre(misNaves);
40  mostrarPorPuntos(misNaves);
41  System.out.println("\nNave con mayor número de puntos: " +
42      mostrarMayorPuntos(misNaves));
43
44  //leer un nombre y mostrar los datos de la nave con dicho nombre,
45  //mensaje de "no encontrado" en caso contrario
46  System.out.println("\nIngrese el nombre de la nave a buscar: ");
47  String nombre= sc.next();
48  int pos = busquedaLinealNombre(misNaves, nombre);
49  if (pos!= -1)
50      System.out.println("Nave encontrada en la posición " + pos);
51  else
52      System.out.println("Nave no encontrada");
53
54  System.out.println("\nOrdenamiento burbuja por puntos: ");
55  ordenarPorPuntosBurbuja(misNaves);
56  mostrarNaves(misNaves);
57
58  System.out.println("\nOrdenamiento burbuja por nombre: ");
59  ordenarPorNombreBurbuja(misNaves);
60  mostrarNaves(misNaves);
61
62  //mostrar los datos de la nave con dicho nombre, mensaje de
63  //"no encontrado" en caso contrario
64  pos=busquedaBinariaNombre(misNaves,nombre);
65  if (pos!=-1)
66      System.out.println("Nave encontrada en la posición " + pos);
67  else
68      System.out.println("Nave no encontrada");
69
70  System.out.println("\nOrdenamiento selección por puntos: ");
71  ordenarPorPuntosSeleccion(misNaves);
72  mostrarNaves(misNaves);
```

```
73
74     System.out.println("\nOrdenamiento selección por nombre: ");
75     ordenarPorNombreSeleccion(misNaves);
76     mostrarNaves(misNaves);
77
78     System.out.println("\nOrdenamiento inserción por puntos: ");
79     ordenarPorPuntosInsercion(misNaves);
80     mostrarNaves(misNaves);
81
82     System.out.println("\nOrdenamiento inserción por nombre: ");
83     ordenarPorNombreInsercion(misNaves);
84     mostrarNaves(misNaves);
85 }
86
87 //Método para mostrar todas las naves
88 public static void mostrarNaves(Nave [] flota){
89     //Usamos un for-each para imprimir
90     for (Nave c : flota) {
91         System.out.println(c);
92     }
93 }
94
95 //Método para mostrar las naves de un nombre ingresado
96 public static void mostrarPorNombre(Nave [] flota){
97     Scanner sc = new Scanner (System.in);
98     System.out.println("\nIngresa un nombre: ");
99     String name = sc.next();
100     for (Nave c: flota) {
101         if (c.getNombre().equals(name))
102             System.out.println(c);
103     }
104 }
105
106 //Método paara mostrar las naves con los puntos inferiores o iguales a
107 //los ingresados
108 public static void mostrarPorPuntos(Nave [] flota){
109     Scanner sc = new Scanner (System.in);
110     System.out.println("\nIngresa los puntos de referencia: ");
111     int point = sc.nextInt();
112     for (Nave c: flota) {
113         if (c.getPuntos() <= point)
114             System.out.println(c);
115     }
116 }
```

```



117
118 //Método que devuelve la Nave con mayor número de Puntos
119 public static Nave mostrarMayorPuntos(Nave [] flota){
120     Nave mayor = flota[0];
121     for (int i = 1; i < flota.length; i++) {
122         if (flota[i].getPuntos() > mayor.getPuntos()) {
123             mayor = flota[i]; // Si hay una nave con más puntos,actualizamos
124         }
125     }
126     return mayor; // Nave con más puntos
127 }
128
129 //Método que busca la nave con un nombre ingresado (Búsqueda lineal)
130 public static int busquedaLinealNombre(Nave[] flota, String nombre) {
131     for (int i = 0; i < flota.length; i++) {
132         if (flota[i].getNombre().equals(nombre))
133             return i;
134     }
135     return -1;
136 }
137
138 //Método que ordena los puntos de menor a mayor (Por burbuja)
139 public static void ordenarPorPuntosBurbuja(Nave[] flota) {
140     for (int i = 0; i < flota.length; i++) {
141         for (int j = 0; j < flota.length - 1; j++) {
142             if (flota[j].getPuntos() > flota[j + 1].getPuntos())
143                 intercambiar(flota, j, j+1);
144         }
145     }
146 }
147
148 //Método que ordena los nombres de A a Z (Por burbuja)
149 public static void ordenarPorNombreBurbuja(Nave[] flota) {
150     for (int i = 0; i < flota.length - 1; i++) {
151         for (int j = 0; j < flota.length - 1 - i; j++) {
152             if (flota[j].getNombre().compareTo(flota[j + 1].getNombre()) > 0)
153                 intercambiar(flota, j, j + 1);
154         }
155     }
156 }
157
158 //Método que intercambia los nombres y puntos de los anteriores métodos
159 private static void intercambiar(Nave[] flota, int i, int j) {
160     String temp = flota[i].getNombre();
161     flota[i].setNombre(flota[j].getNombre());
162     flota[j].setNombre(temp);
163 }

```

```
164
165 //Método que busca la nave con un nombre ingresado (Búsqueda binaria)
166 public static int busquedaBinariaNombre(Nave[] flota, String nombre) {
167     int izquierda = 0;
168     int derecha = flota.length - 1;
169     while (izquierda <= derecha) {
170         int medio = (izquierda + derecha) / 2;
171         String nombreMedio = flota[medio].getNombre();
172         if (nombreMedio.equals(nombre)) {
173             return medio;
174         } else if (nombreMedio.compareTo(nombre) < 0) {
175             izquierda = medio + 1;
176         } else {
177             derecha = medio - 1;
178         }
179     }
180     return -1;
181 }
182
183 //Método que ordena los puntos de menor a mayor (Por selección)
184 public static void ordenarPorPuntosSeleccion(Nave[] flota) {
185     for (int i = 0; i < flota.length - 1; i++) {
186         int indiceMenor = i;
187         for (int j = i + 1; j < flota.length; j++) {
188             if (flota[j].getPuntos() < flota[indiceMenor].getPuntos()) {
189                 indiceMenor = j;
190             }
191         }
192         if (indiceMenor != i) {
193             Nave temp = flota[i];
194             flota[i] = flota[indiceMenor];
195             flota[indiceMenor] = temp;
196         }
197     }
198 }
199
200 //Método que ordena los nombres de A a Z (Por selección)
201 public static void ordenarPorNombreSeleccion(Nave[] flota) {
202     for (int i = 0; i < flota.length - 1; i++) {
203         int indiceMenor = i;
204         for (int j = i + 1; j < flota.length; j++) {
205             if (flota[j].getNombre().compareTo(flota[indiceMenor].getNombre()) < 0)
206                 indiceMenor = j;
207         }
208     }
209     if (indiceMenor != i) {
210         Nave temp = flota[i];
211         flota[i] = flota[indiceMenor];
212         flota[indiceMenor] = temp;
213     }
214 }
215 }
```

```
216
217 //Método que ordena los puntos de menor a mayor (Por inserción)
218 public static void ordenarPorPuntosInsercion(Nave[] flota) {
219     for (int i = 1; i < flota.length; i++) {
220         Nave clave = flota[i];
221         int j = i - 1;
222         while (j >= 0 && flota[j].getPuntos() > clave.getPuntos()) {
223             flota[j + 1] = flota[j];
224             j--;
225         }
226         flota[j + 1] = clave;
227     }
228 }
229
230 //Método que ordena los nombres de A a Z (Por inserción)
231 public static void ordenarPorNombreInsercion(Nave[] flota) {
232     for (int i = 1; i < flota.length; i++) {
233         Nave clave = flota[i];
234         int j = i - 1;
235         while (j >= 0 && flota[j].getNombre().compareTo(clave.getNombre()) > 0) {
236             flota[j + 1] = flota[j];
237             j--;
238         }
239         flota[j + 1] = clave;
240     }
241 }
```



	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 8

## II. PRUEBAS

Ejecución 1 (Nave y DemoBatalla):



- Ingresamos los datos para las 8 naves:

```

run:                               Nave 5
Nave 1                             Nombre: Fallen
Nombre: Amerci                     Fila: 2
Fila: 4                             Columna: 8
Columna: 6                         Estado: true
Estado: true                       Puntos: 29
Puntos: 84                         Nave 6
Nave 2                             Nombre: Dreitus
Nombre: Scrim                       Fila: 8
Fila: 9                             Columna: 2
Columna: 4                         Estado: true
Estado: false                       Puntos: 49
Puntos: 62                         Nave 7
Nave 3                             Nombre: Ocean
Nombre: Mera                        Fila: 7
Fila: 6                             Columna: 1
Columna: 8                         Estado: false
Estado: true                       Puntos: 94
Puntos: 64                         Nave 8
Nave 4                             Nombre: Atlan
Nombre: Donm                        Fila: 7
Fila: 4                             Columna: 5
Columna: 3                         Estado: true
Estado: false                       Puntos: 76
Puntos: 87

```



	<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;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 9</p>

- Mostrar naves creadas:

```
Naves creadas:
Nave:
Nombre->Amerci, Fila->4, Columna->6, Estado->>true, Puntos->84
Nave:
Nombre->Scrim, Fila->9, Columna->4, Estado->>false, Puntos->62
Nave:
Nombre->Mera, Fila->6, Columna->8, Estado->>true, Puntos->64
Nave:
Nombre->Donm, Fila->4, Columna->3, Estado->>false, Puntos->87
Nave:
Nombre->Fallen, Fila->2, Columna->8, Estado->>true, Puntos->29
Nave:
Nombre->Dreitus, Fila->8, Columna->2, Estado->>true, Puntos->49
Nave:
Nombre->Ocean, Fila->7, Columna->1, Estado->>false, Puntos->94
Nave:
Nombre->Atlan, Fila->7, Columna->5, Estado->>true, Puntos->76
```

- Mostrar los datos de un nombre ingresado



```
Ingrese un nombre:
Fallen
Nave:
Nombre->Fallen, Fila->2, Columna->8, Estado->>true, Puntos->29
```

- Mostrar las naves con puntos inferiores o iguales a los ingresados

```
Ingrese los puntos de referencia:
50
Nave:
Nombre->Fallen, Fila->2, Columna->8, Estado->>true, Puntos->29
Nave:
Nombre->Dreitus, Fila->8, Columna->2, Estado->>true, Puntos->49
```

- Muestra la nave con el mayor número de puntos

```
Nave con mayor número de puntos: Nave:
Nombre->Ocean, Fila->7, Columna->1, Estado->>false, Puntos->94
```



	<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;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p style="text-align: center;"><b>Código:</b> GUIA-PRLE-001</p>	<p style="text-align: right;"><b>Página:</b> 10</p>

- Búsqueda lineal de un nombre

```
Ingrese el nombre de la nave a buscar:
Garren
Nave no encontrada
```

- Ordenamiento burbuja por puntos

```
Ordenamiento burbuja por puntos:
Nave:
Nombre->Amerci, Fila->4, Columna->6, Estado->>true, Puntos->84
Nave:
Nombre->Scrim, Fila->9, Columna->4, Estado->>false, Puntos->62
Nave:
Nombre->Mera, Fila->6, Columna->8, Estado->>true, Puntos->64
Nave:
Nombre->Donm, Fila->4, Columna->3, Estado->>false, Puntos->87
Nave:
Nombre->Fallen, Fila->2, Columna->8, Estado->>true, Puntos->29
Nave:
Nombre->Dreitus, Fila->8, Columna->2, Estado->>true, Puntos->49
Nave:
Nombre->Ocean, Fila->7, Columna->1, Estado->>false, Puntos->94
Nave:
Nombre->Atlan, Fila->7, Columna->5, Estado->>true, Puntos->76
```



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

- Ordenamiento burbuja por nombres:

```
Ordenamiento burbuja por nombre:
Nave:
Nombre->Amerci, Fila->4, Columna->6, Estado->>true, Puntos->84
Nave:
Nombre->Atlan, Fila->9, Columna->4, Estado->>false, Puntos->62
Nave:
Nombre->Donm, Fila->6, Columna->8, Estado->>true, Puntos->64
Nave:
Nombre->Dreitus, Fila->4, Columna->3, Estado->>false, Puntos->87
Nave:
Nombre->Fallen, Fila->2, Columna->8, Estado->>true, Puntos->29
Nave:
Nombre->Mera, Fila->8, Columna->2, Estado->>true, Puntos->49
Nave:
Nombre->Ocean, Fila->7, Columna->1, Estado->>false, Puntos->94
Nave:
Nombre->Scrim, Fila->7, Columna->5, Estado->>true, Puntos->76
```

- Ordenamiento selección por puntos

```
Ordenamiento selección por puntos:
Nave:
Nombre->Fallen, Fila->2, Columna->8, Estado->>true, Puntos->29
Nave:
Nombre->Mera, Fila->8, Columna->2, Estado->>true, Puntos->49
Nave:
Nombre->Atlan, Fila->9, Columna->4, Estado->>false, Puntos->62
Nave:
Nombre->Donm, Fila->6, Columna->8, Estado->>true, Puntos->64
Nave:
Nombre->Scrim, Fila->7, Columna->5, Estado->>true, Puntos->76
Nave:
Nombre->Amerci, Fila->4, Columna->6, Estado->>true, Puntos->84
Nave:
Nombre->Dreitus, Fila->4, Columna->3, Estado->>false, Puntos->87
Nave:
Nombre->Ocean, Fila->7, Columna->1, Estado->>false, Puntos->94
```



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

- Ordenamiento selección por nombres:

```
Ordenamiento selección por nombre:
Nave:
Nombre->Amerci, Fila->4, Columna->6, Estado->>true, Puntos->84
Nave:
Nombre->Atlan, Fila->9, Columna->4, Estado->>false, Puntos->62
Nave:
Nombre->Donm, Fila->6, Columna->8, Estado->>true, Puntos->64
Nave:
Nombre->Dreitus, Fila->4, Columna->3, Estado->>false, Puntos->87
Nave:
Nombre->Fallen, Fila->2, Columna->8, Estado->>true, Puntos->29
Nave:
Nombre->Mera, Fila->8, Columna->2, Estado->>true, Puntos->49
Nave:
Nombre->Ocean, Fila->7, Columna->1, Estado->>false, Puntos->94
Nave:
Nombre->Scrim, Fila->7, Columna->5, Estado->>true, Puntos->76
```

- Ordenamiento inserción por puntos

```
Ordenamiento inserción por puntos:
Nave:
Nombre->Fallen, Fila->2, Columna->8, Estado->>true, Puntos->29
Nave:
Nombre->Mera, Fila->8, Columna->2, Estado->>true, Puntos->49
Nave:
Nombre->Atlan, Fila->9, Columna->4, Estado->>false, Puntos->62
Nave:
Nombre->Donm, Fila->6, Columna->8, Estado->>true, Puntos->64
Nave:
Nombre->Scrim, Fila->7, Columna->5, Estado->>true, Puntos->76
Nave:
Nombre->Amerci, Fila->4, Columna->6, Estado->>true, Puntos->84
Nave:
Nombre->Dreitus, Fila->4, Columna->3, Estado->>false, Puntos->87
Nave:
Nombre->Ocean, Fila->7, Columna->1, Estado->>false, Puntos->94
```



	<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;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 13</p>

- Ordenamiento inserción por nombres

```
Ordenamiento inserción por nombre:
Nave:
Nombre->Amerci, Fila->4, Columna->6, Estado->>true, Puntos->84
Nave:
Nombre->Atlan, Fila->9, Columna->4, Estado->>false, Puntos->62
Nave:
Nombre->Donm, Fila->6, Columna->8, Estado->>true, Puntos->64
Nave:
Nombre->Dreitus, Fila->4, Columna->3, Estado->>false, Puntos->87
Nave:
Nombre->Fallen, Fila->2, Columna->8, Estado->>true, Puntos->29
Nave:
Nombre->Mera, Fila->8, Columna->2, Estado->>true, Puntos->49
Nave:
Nombre->Ocean, Fila->7, Columna->1, Estado->>false, Puntos->94
Nave:
Nombre->Scrim, Fila->7, Columna->5, Estado->>true, Puntos->76
BUILD SUCCESSFUL (total time: 5 minutes 22 seconds)
```

## CUESTIONARIO:

- **¿Con que valores comprobaste que tu práctica estuviera correcta?**  
Con Strings, enteros y booleanos. Luego de colocar los datos de las naves como nombres, filas y columna, estado y puntos, ejecutamos el proyecto.  
Durante la ejecución se verifica si el programa está 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 en el main funcionara, que imprimieran, que compararan elementos como para hallar el mayor en un arreglo de objetos, que los métodos ordenaran según burbuja, selección e inserción y que a su vez, las búsquedas funcionaran también.
- **¿Qué valor o comportamiento obtuviste para cada valor de entrada?**  
Se obtuvo como resultado la impresión y muestra de cada valor de entrada. Y según los métodos de búsqueda u ordenamiento, diferentes resultados en los que se concatenó los elementos ingresados.

	<p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<p style="text-align: center;"><b>Formato:</b> 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>

## CONEXIÓN A GITHUB:

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).



```

Directorio: C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src


Mode                LastWriteTime         Length Name
----                -
d-----          13/10/2024    22:44             laboratorio04_karlabedregal

PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git remote add origin https://github.com/KarlaBedregal/Fundamentos2-LABORATORIO.git
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git remote -v
origin  https://github.com/KarlaBedregal/Fundamentos2-LABORATORIO.git (fetch)
origin  https://github.com/KarlaBedregal/Fundamentos2-LABORATORIO.git (push)
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git branch -a
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git checkout master
error: pathspec 'master' did not match any file(s) known to git
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git checkout main
error: pathspec 'main' did not match any file(s) known to git
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git branch -a
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git checkout master
error: pathspec 'master' did not match any file(s) known to git
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git branch -r
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git checkout -b master origin/master
fatal: 'origin/master' is not a commit and a branch 'master' cannot be created from it
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git remote -v
origin  https://github.com/KarlaBedregal/Fundamentos2-LABORATORIO.git (fetch)
origin  https://github.com/KarlaBedregal/Fundamentos2-LABORATORIO.git (push)
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_KarlaBedregal\src> git fetch origin
remote: Enumerating objects: 9, done.
```





	<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;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 15</p>


```
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0 (from 0)
Unpacking objects: 100% (9/9), 3.56 KiB | 191.00 KiB/s, done.
From https://github.com/KarlaBedregal/Fundamentos2-LABORATORIO
* [new branch]      master    -> origin/master
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_Karl
aBedregal\src> git checkout -b master origin/master
branch 'master' set up to track 'origin/master'.
Already on 'master'
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_Karl
aBedregal\src> git branch
* master
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_Karl
aBedregal\src> git add .
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_Karl
aBedregal\src> git commit -m "añadiendo lab 04"
[master 26274bd] añadiendo lab 04
 2 files changed, 290 insertions(+)
 create mode 100644 laboratorio04_karlabedregal/DemoBatalla.java
 create mode 100644 laboratorio04_karlabedregal/Nave.java
PS C:\Users\LENOVO\Documents\NetBeansProjects\Lab01_fp2-KarlaBedregal\LABoratorio04_Karl
aBedregal\src> git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.63 KiB | 2.63 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/KarlaBedregal/Fundamentos2-LABORATORIO.git
ed0fa20..26274bd  master -> master
```



**Fundamentos2-LABORATORIO**
Public
Pin
Unwatch 1
Fork
Star 0


master
Go to file
Code
About


**KarlaBedregal**
añadiendo lab 04
26274bd · 2 minutes ago


laboratorio03\_karlabedregal
Subiendo archivos de lab 03
last week


laboratorio04\_karlabedregal
añadiendo lab 04
2 minutes ago


**README**



## Add a README

Help people interested in this repository understand your project by adding a README.

*No description, website, or topics provided.*

Activity
0 stars
1 watching
0 forks



### Releases

No releases published

[Create a new release](#)

### Packages



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



## 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 investigado los diferentes tipos de ordenamiento y su funcionamiento, 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.

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 17

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	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		18	

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 18</p>

## REFERENCIAS Y BIBLIOGRAFÍA

<https://github.com/LINOPINTO2023/FundProg2/tree/main/IEEE>