



|  |  |   |
|--|--|---|
|   | <p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN<br/>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS<br/>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:                                  | Laboratorio Fundamentos de la Programación 2   |                      |          |                |                              |
| TÍTULO DE LA PRÁCTICA:                       | Arreglos de Objetos, Búsquedas y Ordenamientos |                      |          |                |                              |
| NÚMERO DE PRÁCTICA:                          | 04   | AÑO LECTIVO:         | 2024     | NRO. SEMESTRE: | 02                           |
| FECHA DE PRESENTACIÓN                        | 15/10/2024                                     | HORA DE PRESENTACIÓN | 03/20/45 |                |                              |
| INTEGRANTE (s)<br>Sergio Emilio Estrada Arce |  |                      |          | NOTA (0-20)    | Nota colocada por el docente |
| DOCENTE(s):<br>Ing. Lino José Pinto Oppe     |  |                      |          |                |                              |

| RESULTADOS Y PRUEBAS                                       |
|--|
| <p><b>I. EJERCICIOS RESUELTOS:</b></p> <p>Ejercicio 1:</p> |

```
1 //Laboratorio Nro 4 - Ejercicio 1
2 //Autor: Sergio Estrada Arce
3 //Tiempo: 5h 39m
4 package Ejercicios;
5 import java.util.*;
6 public class DemoBatalla {
7     public static void main(String[] args) {
8         Nave[] misNaves = new Nave[8];
9         Scanner sc = new Scanner(System.in);
10        String nomb, col;
11        int fil, punt;
12        boolean est;
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.print("Fila: ");
18            fil = sc.nextInt();
19            System.out.print("Columna: ");
20            col = sc.next();
21            System.out.print("Estado (true/false): ");
22            est = sc.nextBoolean();
23            System.out.print("Puntos: ");
24            punt = sc.nextInt();
25            misNaves[i] = new Nave();
26            misNaves[i].setNombre(nomb);
27            misNaves[i].setFila(fil);
28            misNaves[i].setColumna(col);
29            misNaves[i].setEstado(est);
30            misNaves[i].setPuntos(punt);
31        }
32        // Mostrar todas las naves
33        System.out.println("\nNaves creadas:");
34        mostrarNaves(misNaves);
35        // Mostrar naves por nombre
36        mostrarPorNombre(misNaves);
37        // Mostrar naves por puntos
38        mostrarPorPuntos(misNaves);
39        // Nave con mayor número de puntos
40        System.out.println("\nNave con mayor número de puntos: " + mostrarMayorPuntos(misNaves).getNombre());
41        // Buscar nave por nombre (búsqueda lineal)
42        System.out.print("\nIngrese un nombre para buscar (búsqueda lineal): ");
43        String nombre = sc.next();
44        int pos = busquedaLinealNombre(misNaves, nombre);
45        if (pos != -1) {
46            System.out.println("Nave encontrada: " + misNaves[pos].getNombre());
47        } else {
48            System.out.println("Nave no encontrada.");
49        }
50        // Ordenar naves por puntos (burbuja)
51        ordenarPorPuntosBurbuja(misNaves);
52        System.out.println("\nNaves ordenadas por puntos (de menor a mayor) - Burbuja:");
53        mostrarNaves(misNaves);
54        // Ordenar naves por nombre (burbuja)
55        ordenarPorNombreBurbuja(misNaves);
56        System.out.println("\nNaves ordenadas por nombre (A a Z) - Burbuja:");
57        mostrarNaves(misNaves);
58        // Buscar nave por nombre (búsqueda binaria)
59        ordenarPorNombreBurbuja(misNaves); // Asegurarse de que las naves están ordenadas
60        System.out.print("\nIngrese un nombre para buscar (búsqueda binaria): ");
```

```



61     nombre = sc.next();
62     pos = busquedaBinariaNombre(misNaves, nombre);
63     if (pos != -1) {
64         System.out.println("Nave encontrada: " + misNaves[pos].getNombre());
65     } else {
66         System.out.println("Nave no encontrada.");
67     }
68     // Ordenar naves por puntos (selección)
69     ordenarPorPuntosSelección(misNaves);
70     System.out.println("\nNaves ordenadas por puntos (de menor a mayor) - Selección:");
71     mostrarNaves(misNaves);
72     // Ordenar naves por nombre (selección)
73     ordenarPorNombreSelección(misNaves);
74     System.out.println("\nNaves ordenadas por nombre (A a Z) - Selección:");
75     mostrarNaves(misNaves);
76     // Ordenar naves por puntos (inserción)
77     ordenarPorPuntosInserción(misNaves);
78     System.out.println("\nNaves ordenadas por puntos (de mayor a menor) - Inserción:");
79     mostrarNaves(misNaves);
80     // Ordenar naves por nombre (Z a A) - Inserción
81     ordenarPorNombreInserción(misNaves);
82     System.out.println("\nNaves ordenadas por nombre (Z a A) - Inserción:");
83     mostrarNaves(misNaves);
84 }
85 // Método para mostrar todas las naves
86 public static void mostrarNaves(Nave[] flota) {
87     for (Nave nave : flota) {
88         System.out.println(nave);
89     }
90 }
91 // Método para mostrar naves por nombre pedido por teclado
92 public static void mostrarPorNombre(Nave[] flota) {
93     Scanner sc = new Scanner(System.in);
94     System.out.print("\nIngrese el nombre de la nave a buscar: ");
95     String nombre = sc.next();
96     boolean encontrado = false;
97     for (Nave nave : flota) {
98         if (nave.getNombre().equals(nombre)) {
99             System.out.println(nave);
100             encontrado = true;
101         }
102     }
103     if (!encontrado) {
104         System.out.println("No se encontró ninguna nave con ese nombre.");
105     }
106 }
107 // Método para mostrar naves con puntos menores o iguales al pedido
108 public static void mostrarPorPuntos(Nave[] flota) {
109     Scanner sc = new Scanner(System.in);
110     System.out.print("\nIngrese un número de puntos para buscar naves con puntos menores o iguales: ");
111     int puntos = sc.nextInt();
112     for (Nave nave : flota) {
113         if (nave.getPuntos() <= puntos) {
114             System.out.println(nave);
115         }
116     }
117 }
118 // Método que devuelve la nave con mayor número de puntos
119 public static Nave mostrarMayorPuntos(Nave[] flota) {
120     Nave naveMayor = flota[0];

```

```

121     for (Nave nave : flota) {
122         if (nave.getPuntos() > naveMayor.getPuntos()) {
123             naveMayor = nave;
124         }
125     }
126     return naveMayor;
127 }
128 // Método para buscar la primera nave con un nombre (búsqueda lineal)
129 public static int busquedaLinealNombre(Nave[] flota, String s) {
130     for (int i = 0; i < flota.length; i++) {
131         if (flota[i].getNombre().equals(s)) {
132             return i;
133         }
134     }
135     return -1;
136 }
137 // Método que ordena por puntos (de menor a mayor) usando Burbuja
138 public static void ordenarPorPuntosBurbuja(Nave[] flota) {
139     for (int i = 0; i < flota.length - 1; i++) {
140         for (int j = 0; j < flota.length - 1 - i; j++) {
141             if (flota[j].getPuntos() > flota[j + 1].getPuntos()) {
142                 Nave temp = flota[j];
143                 flota[j] = flota[j + 1];
144                 flota[j + 1] = temp;
145             }
146         }
147     }
148 }
149 // Método que ordena por nombre (A a Z) usando Burbuja
150 public static void ordenarPorNombreBurbuja(Nave[] flota) {
151     for (int i = 0; i < flota.length - 1; i++) {
152         for (int j = 0; j < flota.length - 1 - i; j++) {
153             if (flota[j].getNombre().compareTo(flota[j + 1].getNombre()) > 0) {
154                 Nave temp = flota[j];
155                 flota[j] = flota[j + 1];
156                 flota[j + 1] = temp;
157             }
158         }
159     }
160 }
161 // Método que ordena por puntos (de menor a mayor) usando Selección
162 public static void ordenarPorPuntosSeleccion(Nave[] flota) {
163     for (int i = 0; i < flota.length - 1; i++) {
164         int minIndex = i;
165         for (int j = i + 1; j < flota.length; j++) {
166             if (flota[j].getPuntos() < flota[minIndex].getPuntos()) {
167                 minIndex = j;
168             }
169         }
170         Nave temp = flota[i];
171         flota[i] = flota[minIndex];
172         flota[minIndex] = temp;
173     }
174 }
175 // Método que ordena por nombre (A a Z) usando Selección
176 public static void ordenarPorNombreSeleccion(Nave[] flota) {
177     for (int i = 0; i < flota.length - 1; i++) {
178         int minIndex = i;
179         for (int j = i + 1; j < flota.length; j++) {
180             if (flota[j].getNombre().compareTo(flota[minIndex].getNombre()) < 0) {

```

|   |   |   |
|---|---|---|
|                                      | <p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b><br/> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b><br/> <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> 5</p>   |



```

181         minIndex = j;
182     }
183 }
184 Nave temp = flota[i];
185 flota[i] = flota[minIndex];
186 flota[minIndex] = temp;
187 }
188 }
189 // Método que ordena por puntos (de mayor a menor) usando Inserción
190 public static void ordenarPorPuntosInsercion(Nave[] flota) {
191     for (int i = 1; i < flota.length; i++) {
192         Nave key = flota[i];
193         int j = i - 1;
194         while (j >= 0 && flota[j].getPuntos() < key.getPuntos()) {
195             flota[j + 1] = flota[j];
196             j--;
197         }
198         flota[j + 1] = key;
199     }
200 }
201
202 // Método que ordena por nombre (Z a A) usando Inserción
203 public static void ordenarPorNombreInsercion(Nave[] flota) {
204     for (int i = 1; i < flota.length; i++) {
205         Nave key = flota[i];
206         int j = i - 1;
207         while (j >= 0 && flota[j].getNombre().compareTo(key.getNombre()) < 0) {
208             flota[j + 1] = flota[j];
209             j--;
210         }
211         flota[j + 1] = key;
212     }
213 }
214 // Método para buscar una nave por nombre (búsqueda binaria)
215 public static int busquedaBinariaNombre(Nave[] flota, String s) {
216     int inicio = 0;
217     int fin = flota.length - 1;
218
219     while (inicio <= fin) {
220         int mid = (inicio + fin) / 2;
221         if (flota[mid].getNombre().equals(s)) {
222             return mid;
223         } else if (flota[mid].getNombre().compareTo(s) < 0) {
224             inicio = mid + 1;
225         } else {
226             fin = mid - 1;
227         }
228     }
229     return -1;
230 }
231 }
232

```

## II. PRUEBAS

¿Con que valores comprobaste? Utilicé varios conjuntos de valores de prueba, incluyendo nombres como "Spiral", "Hymperiom", y "Perú", filas como 1, 4 y 3, columnas como "J", "B" y "F", estados booleanos (true/false), y puntos enteros como 92, 12, y 45. ¿que tu práctica estuviera correcta?

|   |   |   |
|---|---|---|
|                                      | <p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b><br/> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b><br/> <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> 6</p>   |



*¿Qué resultado esperabas obtener para cada valor de entrada?*

*Esperaba que cada nave se creara correctamente y que, al buscar por nombre, se recuperaran las naves correspondientes. Además, al ordenar por puntos, esperaba que las naves se clasificaran de menor a mayor, y que la búsqueda de un nombre específico devolviera la nave correcta.*

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

*El comportamiento del programa fue el esperado: todas las naves se crearon correctamente con los atributos ingresados. Las búsquedas por nombre devolvieron los objetos correctos, y los métodos de ordenamiento funcionaron adecuadamente, mostrando las naves en el orden correcto según los puntos y nombres.*

*Verificación:*

|   |  |   |
|---|--|---|
|                                      | <p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN<br/>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS<br/>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> 7</p>   |

```

Nave 1
Nombre: Spiral
Fila:
1
Columna: B
Estado (true/false): true
Puntos: 92
Nave 2
Nombre: Hymperion
Fila: 3
Columna: F
Estado (true/false): true
Puntos: 12
Nave 3
Nombre: Perú
Fila: 4
Columna: J
Estado (true/false): true
Puntos: 45
Nave 4
Nombre: Chile
Fila: 2
Columna: K
Estado (true/false): true
Puntos: 34
Nave 5
Nombre: Huascar
Fila: 3
Columna: P
Estado (true/false): true
Puntos: 63
Nave 6
Nombre: Augustogrado
Fila: 7
Columna: Q
Estado (true/false): true
Puntos: 98
Nave 7
Nombre: Fring
Fila: 3
Columna: H
Estado (true/false): false
Puntos: 54
Nave 8
Nombre: Estrada
Fila: 5
Columna: U
Estado (true/false): true
Puntos: 123

```

Naves creadas:

```

Nave{nombre='Spiral', fila=1, columna='B', estado=true, puntos=92}
Nave{nombre='Hymperion', fila=3, columna='F', estado=true, puntos=12}
Nave{nombre='perú', fila=4, columna='J', estado=true, puntos=45}
Nave{nombre='Chile', fila=2, columna='K', estado=true, puntos=34}
Nave{nombre='Huascar', fila=3, columna='P', estado=true, puntos=63}
Nave{nombre='Augustogrado', fila=7, columna='Q', estado=true, puntos=98}
Nave{nombre='Fring', fila=3, columna='H', estado=false, puntos=54}
Nave{nombre='Estrada', fila=5, columna='U', estado=true, puntos=123}

```



Ingrese el nombre de la nave a buscar: **Chile**

Nave{nombre='Chile', fila=2, columna='K', estado=true, puntos=34}

Ingrese un número de puntos para buscar naves con puntos menores o iguales: **6:**

Nave{nombre='Hymperion', fila=3, columna='F', estado=true, puntos=12}

Nave{nombre='perú', fila=4, columna='J', estado=true, puntos=45}

Nave{nombre='Chile', fila=2, columna='K', estado=true, puntos=34}

Nave{nombre='Huascar', fila=3, columna='P', estado=true, puntos=63}

Nave{nombre='Fring', fila=3, columna='H', estado=false, puntos=54}

Nave con mayor número de puntos: Estrada

Ingrese un nombre para buscar (búsqueda lineal): **Augustogrado**

Nave encontrada: Augustogrado

Naves ordenadas por puntos (de menor a mayor) - Burbuja:

Nave{nombre='Hymperion', fila=3, columna='F', estado=true, puntos=12}

Nave{nombre='Chile', fila=2, columna='K', estado=true, puntos=34}

Nave{nombre='perú', fila=4, columna='J', estado=true, puntos=45}

Nave{nombre='Fring', fila=3, columna='H', estado=false, puntos=54}

Nave{nombre='Huascar', fila=3, columna='P', estado=true, puntos=63}

Nave{nombre='Spiral', fila=1, columna='B', estado=true, puntos=92}

Nave{nombre='Augustogrado', fila=7, columna='Q', estado=true, puntos=98}

Nave{nombre='Estrada', fila=5, columna='U', estado=true, puntos=123}

Naves ordenadas por nombre (A a Z) - Burbuja:

Nave{nombre='Augustogrado', fila=7, columna='Q', estado=true, puntos=98}

Nave{nombre='Chile', fila=2, columna='K', estado=true, puntos=34}

Nave{nombre='Estrada', fila=5, columna='U', estado=true, puntos=123}

Nave{nombre='Fring', fila=3, columna='H', estado=false, puntos=54}

Nave{nombre='Huascar', fila=3, columna='P', estado=true, puntos=63}

Nave{nombre='Hymperion', fila=3, columna='F', estado=true, puntos=12}

Nave{nombre='Spiral', fila=1, columna='B', estado=true, puntos=92}

Nave{nombre='perú', fila=4, columna='J', estado=true, puntos=45}

Ingrese un nombre para buscar (búsqueda binaria): **Hymperion**

Nave encontrada: Hymperion

Naves ordenadas por puntos (de menor a mayor) - Selección:

Nave{nombre='Hymperion', fila=3, columna='F', estado=true, puntos=12}

Nave{nombre='Chile', fila=2, columna='K', estado=true, puntos=34}

Nave{nombre='perú', fila=4, columna='J', estado=true, puntos=45}

Nave{nombre='Fring', fila=3, columna='H', estado=false, puntos=54}

Nave{nombre='Huascar', fila=3, columna='P', estado=true, puntos=63}

Nave{nombre='Spiral', fila=1, columna='B', estado=true, puntos=92}

Nave{nombre='Augustogrado', fila=7, columna='Q', estado=true, puntos=98}

Nave{nombre='Estrada', fila=5, columna='U', estado=true, puntos=123}

Naves ordenadas por nombre (A a Z) - Selección:

Nave{nombre='Augustogrado', fila=7, columna='Q', estado=true, puntos=98}

Nave{nombre='Chile', fila=2, columna='K', estado=true, puntos=34}

Nave{nombre='Estrada', fila=5, columna='U', estado=true, puntos=123}

Nave{nombre='Fring', fila=3, columna='H', estado=false, puntos=54}



Nave{nombre='Huascar', fila=3, columna='P', estado=true, puntos=63}

Nave{nombre='Hymperion', fila=3, columna='F', estado=true, puntos=12}

Nave{nombre='Spiral', fila=1, columna='B', estado=true, puntos=92}

Nave{nombre='perú', fila=4, columna='J', estado=true, puntos=45}



|   |   |   |
|---|---|---|
|                                      | <p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b><br/> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b><br/> <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> 9</p>   |

Naves ordenadas por puntos (de mayor a menor) - Inserción:

```
Nave{nombre='Estrada', fila=5, columna='U', estado=true, puntos=123}
Nave{nombre='Augustogrado', fila=7, columna='Q', estado=true, puntos=98}
Nave{nombre='Spiral', fila=1, columna='B', estado=true, puntos=92}
Nave{nombre='Huascar', fila=3, columna='P', estado=true, puntos=63}
Nave{nombre='Fring', fila=3, columna='H', estado=false, puntos=54}
Nave{nombre='perú', fila=4, columna='J', estado=true, puntos=45}
Nave{nombre='Chile', fila=2, columna='K', estado=true, puntos=34}
Nave{nombre='Hymperion', fila=3, columna='F', estado=true, puntos=12}
```

Naves ordenadas por nombre (Z a A) - Inserción:

```
Nave{nombre='perú', fila=4, columna='J', estado=true, puntos=45}
Nave{nombre='Spiral', fila=1, columna='B', estado=true, puntos=92}
Nave{nombre='Hymperion', fila=3, columna='F', estado=true, puntos=12}
Nave{nombre='Huascar', fila=3, columna='P', estado=true, puntos=63}
Nave{nombre='Fring', fila=3, columna='H', estado=false, puntos=54}
Nave{nombre='Estrada', fila=5, columna='U', estado=true, puntos=123}
Nave{nombre='Chile', fila=2, columna='K', estado=true, puntos=34}
Nave{nombre='Augustogrado', fila=7, columna='Q', estado=true, puntos=98}
```

Commits:

Commits

main

All users

All time

Commits on Oct 15, 2024

Add files via upload

SrEstrada authored now

Verified

a576afd

<>

Add files via upload

SrEstrada authored 2 minutes ago

Verified

91f5788

<>

Commits on Oct 14, 2024

Add files via upload

SrEstrada authored 4 hours ago

Verified

4db538a

<>

Add files via upload

SrEstrada authored 4 hours ago

Verified

aced5e8

<>

Delete FP2 lab/Lab4/ClaseNave/DemoBatalla.java

SrEstrada authored 4 hours ago

Verified

1cd0878

<>

Add files via upload

SrEstrada authored 4 hours ago

Verified

41f2825

<>

Commits on Oct 11, 2024

Create Clase nave

SrEstrada authored 3 days ago

Verified

25e7629

<>

Add files via upload

SrEstrada authored 3 days ago

Verified

06022da

<>

Create DemoBatallaVersión1

SrEstrada authored 3 days ago

Verified

44611d3

<>

Delete FP2 lab/Lab4/Main Primera versión

SrEstrada authored 3 days ago

Verified

4a29c56

<>

Create Main Primera versión

SrEstrada authored 3 days ago

Verified

36e853a

<>

Add files via upload

SrEstrada authored 3 days ago

Verified



25174b8

<>

III. CUESTIONARIO:

CONCLUSIONES

Lo que hice en si fue implementar correctamente los métodos para manipular la clase Nave, aplicando algoritmos de ordenamiento y búsqueda. Los resultados obtenidos coincidieron con lo que queria, demostrando el funcionamiento adecuado del código. Esta me hace darme cuenta de la importancia de la modularidad y la reutilización de código en el desarrollo de software.

|   |  |   |
|---|--|---|
|          | <p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN<br/>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS<br/>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p> |  |
| <p><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> 11</p>  |



### METODOLOGÍA DE TRABAJO

*Primero, analicé el código base y determiné las áreas donde se podían optimizar los métodos. Luego, implementé las mejoras, reutilizando funciones y aplicando algoritmos eficientes de ordenamiento y búsqueda. Finalmente, realicé pruebas para asegurar que cada método cumpliera con los requisitos establecidos.*



### REFERENCIAS Y BIBLIOGRAFÍA

GitHub - SrEstrada/Laboratorios\_Estrada\_Arce. (s.f.).

GitHub. [https://github.com/SrEstrada/Laboratorios\\_Estrada\\_Arce.git](https://github.com/SrEstrada/Laboratorios_Estrada_Arce.git)

|  |   |   |
|--|---|---|
|   | <p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b><br/> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b><br/> <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: 12  |

| 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.<br><br>(El profesor puede preguntar para refrendar calificación). | 4      | X         | 3          |          |
| 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         | 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          |          |
| 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                              | 4      | X         | 3          |          |

|   |   |   |
|---|---|---|
|                                      | <p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b><br/> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b><br/> <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> 13</p>  |

|  |  |                                       |  |                                       |  |
|--|--|---------------------------------------|--|---------------------------------------|--|
|  | <p>acabado impecable. (El profesor puede preguntar para refrendar calificación).</p> |                                       |  |                                       |  |
| <p style="text-align: center;">TOTAL</p> |  | <p style="text-align: center;">20</p> |  | <p style="text-align: center;">16</p> |  |