
	<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:	Fundamentos de la programación 2				
TÍTULO DE LA PRÁCTICA:	Arreglos estándar				
NÚMERO DE PRÁCTICA:	02	AÑO LECTIVO:	2024	NRO. SEMESTRE:	II
FECHA DE PRESENTACIÓN	28/09/2024	HORA DE PRESENTACIÓN	10:50:00		
INTEGRANTE (s) Riveros Vilca Alberth Edwar				NOTA (0-20)	
DOCENTE(s): Ing. Lino José Pinto Oppe					

RESULTADOS Y PRUEBAS
<p><b>I. EJERCICIOS RESUELTOS:</b></p> <p><i>El estudiante coloca la evidencia de los ejercicios propuestos realizados en la sesión de laboratorio, en el tiempo o duración indicado por el docente.</i></p> <p><i>El docente debe colocar la retroalimentación por cada ejercicio que el estudiante/grupo ha presentado</i></p> <p><b>JUEGO DEL AHORCADO</b></p> <p><i>En este ejercicio se le solicita a usted implementar el juego del ahorcado utilizando el código parcial que se le entrega.</i></p> <p><i>Deberá considerar que:</i></p> <ul style="list-style-type: none"> <li>• El juego valida el ingreso de letras solamente. En caso el usuario ingrese un carácter equivocado le dará el mensaje de error y volverá a solicitar el ingreso</li> <li>• El juego supone que el usuario no ingresa una letra ingresada previamente</li> <li>• El método <code>ingreseLetra()</code> debe ser modificado para incluir las consideraciones de validación</li> <li>• Puede crear métodos adicionales</li> </ul>

```
1 import java.util.Scanner;
2 public class JuegoAhorcado { new *
3     public static void main(String []args) { new *
4         String ahor1 = ""
5             +---+
6             |   |
7             |   |
8             |   |
9             |   |
10            |   |
11            =====;
12        String ahor2 = ""
13            +---+
14            |   |
15            0   |
16            |   |
17            |   |
18            |   |
19            =====;
20        String ahor3 = ""
21            +---+
22            |   |
23            0   |
24            |   |
25            |   |
26            |   |
27            =====;
28        String ahor4 = ""
29            +---+
30            |   |
31            0   |
32            /|  |
33            |   |
34            |   |
35            =====;
36        String ahor5 = ""
37            +---+
38            |   |
39            0   |
40            /|\  |
41            |   |
42            |   |
```

```

43         ===== "  ";
44     String ahor6 = "  "
45         +---+
46         |  |
47         0  |
48         /|\ |
49         /  |
50         |
51         ===== "  ";
52     String ahor7 = "  "
53         +---+
54         |  |
55         0  |
56         /|\ |
57         / \ |
58         |
59         ===== "  ";
60     String[] figuras = {ahor1, ahor2, ahor3, ahor4, ahor5, ahor6, ahor7};
61     int errores = 0;
62     int intentosEmpleados = 0;
63     char letra;
64     String adivinado = "";
65     String[] palabras = {"programacion", "java", "indentacion", "clases",
66         "objetos", "desarrollador", "pruebas"};
67     String palSecreta = getPalabraSecreta(palabras);
68     for (int i = 0; i < palSecreta.length(); i++) {
69         adivinado += "-";
70     }
71     System.out.println(figuras[0]);
72     mostrarBlancos(palSecreta);
73     System.out.println("\n");
74     /*Verifica que aún no haya adivinado la palabra y este dentro de los
75     errores permitidos para el bucle while*/
76     while (errores < 6 && !adivinado.equals(palSecreta)) {
77         letra = ingreseLetra();
78         intentosEmpleados++;
79         if (letraEnPalabraSecreta(letra, palSecreta)) {
80             adivinado = mostrarBlancosActualizados(letra, palSecreta, adivinado);
81             System.out.println("Acertaste una letra :D");



```

```

82         System.out.println(advertido);
83     } else {
84         System.out.println(advertidos[errores+1]);
85         errores++;
86         System.out.println(advertido);
87         System.out.println("No acertaste :c");
88         System.out.println("Intentos restantes: " + (6- errores) + "\n");
89     }
90 }
91 //Verificamos si se adivino la palabra e imprimimos el mensaje correspondiente
92 if(advertido.equals(palSecreta)){
93     System.out.println("¡Usted ganó en " + intentosEmpleados + " intentos!");
94 }else{
95     System.out.println("Usted perdió. La palabra era: " + palSecreta);
96 }
97 }
98 @
99 public static String getPalabraSecreta(String [] lasPalabras){ 1usage new *
100     /*Se escoge aleatoriamente la palabra Secreta de todas las disponibles*/
101     int ind = (int) (Math.random() * (lasPalabras.length));
102     return lasPalabras[ind];
103 }
104 @
105 public static void mostrarBlancos(String palabra) { 1usage new *
106     /*Muestra los blancos de la palabra secreta*/
107     System.out.println("Cual será la palabra secreta?");
108     for(int i=0; i< palabra.length(); i++)
109         System.out.print("_");
110 }
111 public static char ingreseLetra(){ 1usage new *
112     /*Se ingresa un caracter para el juego*/
113     Scanner sc = new Scanner(System.in);
114     System.out.println("Ingrese letra: ");
115     String laLetra = sc.next().toLowerCase();
116     //Verifica que solo sea un caracter del rango a-z
117     while(laLetra.length()!= 1 || !Character.isLetter(laLetra.charAt(0))) {
118         System.out.println("Dato invalido,por favor ingrese un caracter en el rango a-z: ");
119         laLetra = sc.next().toLowerCase();
120     }
121     return laLetra.charAt(0);
122 }
123 @
124 public static boolean letraEnPalabraSecreta(char letra, String palSecreta){ 1usage new *
125     /*Verifica si el caracter se encuentra en la palabra secreta*/
126     for(int i=0;i<palSecreta.length();i++){
127         if(palSecreta.charAt(i)==letra) {
128             return true;
129         }
130     }
131     return false;
132 }
133 @
134 public static String mostrarBlancosActualizados(char letra,String palSecreta,String palAdivinada){ 1usage new *
135     /*Aqui se actualiza los blancos de la palabra*/
136     System.out.println("PROCESANDO....");
137     //Se utiliza un arreglo de caracteres de la palabra Adivinada
138     char[] palAdivinadoChars = palAdivinada.toCharArray();
139     for(int i=0;i<palSecreta.length();i++){//Actualiza cada caracter en el arreglo
140         if(palSecreta.charAt(i)==letra)
141             palAdivinadoChars[i] = letra;
142     }
143     return String.valueOf(palAdivinadoChars);
144 }
145 }

```

*El programa verifica que los datos ingresados sean caracteres en el rango a-z.*

	<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: 6</p>

*Si el usuario erra mostrara la figura y un mensaje informando sobre los intentos restantes.*

```

Ingrese letra:
ñ
+---+
|   |
0   |
|   |
|   |
|   |
=====
-----
No acertaste :c
Intentos restantes: 4

```

*Al final si pierde mostrara la palabra secreta.*

```

Ingrese letra:
w
+---+
|   |
0   |
/ \ |
/ \ |
|   |
=====
-----
No acertaste :c
Intentos restantes: 0

Usted perdió. La palabra era: clases
Process finished with exit code 0

```

*Y en el caso contrario el número de intentos necesarios para encontrar la palabra.*

```

¡Usted ganó en 11 intentos!

Process finished with exit code 0



```

### III. PRUEBAS

*¿Con que valores comprobaste que tu práctica estuviera correcta?*

*Lo compruebe, con string, números y caracteres para verificar correctamente las entradas válidas.*

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

	<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> 7</p>

Para el string y números esperaba un mensaje de entrada invalidad y que me requiera otra entrada,y para los caracteres que verificara si estaba en la palabra secreta en caso de que fuera cierto mostrab la palabra actualizada y en caso contrario la figura correspondiente y los intentos restantes.

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

Obtuve lo que espere pero tuve un error con las figuras correspondientes y el intento,lo solucione corrigiendo los indices y valores iniciales de la variable errores.

## REPOSITORIO:

Inicializando el repositorio:

```
~/IdeaProjects/Laboratorio_02/src
git init
hint: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
hint: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
hint: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
hint:
hint: git config --global init.defaultBranch <nombre>
hint:
hint: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
hint: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este comando:
hint:
hint: git branch -m <nombre>
Iniciado repositorio Git vacío en /home/riv/IdeaProjects/Laboratorio_02/src/.git/
```

Agregando el archivo README.md y realizamos el primer commit

```
~/IdeaProjects/Laboratorio_02/src on git P master ?2
git add README.md

~/IdeaProjects/Laboratorio_02/src on git P master +1 ?1
git commit -m "first commit"
[master (commit-raiz) 1901461] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Cambiamos el nombre de la rama a main

```
~/IdeaProjects/Laboratorio_02/src on git P master ?1
git branch -M main
```

Añadimos el remoto origin

```
~/IdeaProjects/Laboratorio_02/src on git P main ?1
git remote add origin https://github.com/rivX241/RIVEROS_VILCA_LABORATORIO_02.git
```

Envío los commits a origin el repositorio en Github

```
~/IdeaProjects/Laboratorio_02/src on git P main ?1
git push -u origin main
Username for 'https://github.com': rivX241
Password for 'https://rivX241@github.com':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 243 bytes | 243.00 KiB/s, listo.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/rivX241/RIVEROS_VILCA_LABORATORIO_02.git
 * [new branch] main -> main
rama 'main' configurada para rastrear 'origin/main'.
```

Añado el archivo del ejercicio para el siguiente commit

```
~/IdeaProjects/Laboratorio_02/src on git P main ?1
git add JuegoAhorcado.java
```

Creo el segundo commit

```
~/IdeaProjects/Laboratorio_02/src on git P main +1
git commit -m "second commit"
[main ef6d856] second commit
1 file changed, 141 insertions(+)
create mode 100644 JuegoAhorcado.java
```

Envío el segundo commit



```



~/IdeaProjects/Laboratorio_02/src on es main +1
git push -u origin main
Username for 'https://github.com': rivX241
Password for 'https://rivX241@github.com':
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 1.58 KiB | 1.58 MiB/s, listo.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/rivX241/RIVEROS_VILCA_LABORATORIO_02.git
1901461..ef6d856 main -> main
rama 'main' configurada para rastrear 'origin/main'.
  
```

### III. RUBRICA:

*Colocar la evidencia de las respuestas realizadas al cuestionario enunciado en la guía práctica de laboratorio.*

Contenido y demostración		Puntos	Checklis t	Estudiant e	Profe 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		2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4		4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2		2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2		2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2		2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2		2	
7. Ortografía	El documento no muestra errores ortográficos.	2		1	
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		3	
TOTAL		20		18	



	<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> 9</p>

*Gracias al juego de ahorcados he podido comprender la capacidad de los arreglos de trabajar con muchos datos, y como estos se pueden emplear de manera mucho más efectivas que variables independientes para el uso de bucles y entradas, además me permitió tener un código mucho más limpio y legible que me permitió solucionar el problema.*

## METODOLOGÍA DE TRABAJO

- Analice el programa que me brindaron, y analice el código viendo lo que trataba de hacer, repase la lógica necesaria detrás del juego del ahorcado para así poder comprender los flujos de control y bucles necesarios para el juego.*
- Complete los métodos incompletos y adapte algunos de acuerdo a mi propuesta de solución.*
- Puse a prueba mi solución con las entradas que se podrían ingresar en el peor de los casos y probando los escenarios donde se adivinaba o no la palabra secreta.*
- Documente con comentarios el código para llevar registro de como está estructurado y como funciona cada método.*

## REFERENCIAS Y BIBLIOGRAFÍA

*E. G. Castro Gutiérrez y M. W. Aedo López, Fundamentos de programación 2: tópicos de programación orientada a objetos, 1.ª ed. Universidad Nacional de San Agustín, 2021. ISBN: 978-612-5035-20-2.*

*<https://stackoverflow.com/questions/10006165/convert-string-to-character-array-in-java>*