
	<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: 1

INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	<i>Fundamentos de la programación 2</i>				
TÍTULO DE LA PRÁCTICA:					
NÚMERO DE PRÁCTICA:	<i>02</i>	AÑO LECTIVO:	<i>2024</i>	NRO. SEMESTRE:	<i>//</i>
FECHA DE PRESENTACIÓN	<i>26/09/2024</i>	HORA DE PRESENTACIÓN	<i>10:51:00</i>		
INTEGRANTE (s) <i>Rodrigo Fabricio Layme Salas</i>				NOTA (0-20)	
DOCENTE(s): <i>Ing. Lino José Pinto Oppe</i>					

RESULTADOS Y PRUEBAS
<p>I. EJERCICIOS RESUELTOS:</p> <p>JUEGO DEL AHORCADO:</p> <p>CÓDIGO:</p> <pre> 1 /*PROPOSITO: RECREAR EL JUEGO DEL AHORCADO 2 * AUTOR: LAYME SALAS RODRIGO FABRICIO */ 3 import java.util.*; 4 public class E1 { 5 Run Debug 6 public static void main(String[] args) { 7 String ahor1 = " +---+ \n" + 8 " \n" + 9 " \n" + 10 " \n" + 11 " \n" + 12 "===== "; 13 String ahor2 = " +---+ \n" + 14 " \n" + 15 " 0 \n" + 16 " \n" + 17 " \n" + 18 "===== "; 19 String ahor3 = " +---+ \n" + 20 " \n" + 21 " 0 \n" + 22 " \n" + 23 " \n" + 24 "===== "; 25 String ahor4 = " +---+ \n" + 26 " \n" + 27 " 0 \n" + 28 " / \n" + 29 " \n" + 30 "===== "; 31 String ahor5 = " +---+ \n" + 32 " \n" + 33 " 0 \n" + 34 " / \ \n" + 35 " \n" + 36 "===== "; 37 </pre>

```

39         "         | \n" +
40         "=====";
41     String ahor6 = " +---+ \n" +
42         " | | \n" +
43         " 0 | \n" +
44         " /\\ | \n" +
45         " /  | \n" +
46         " | | \n" +
47         "=====";
48     String ahor7 = " +---+ \n" +
49         " | | \n" +
50         " 0 | \n" +
51         " /\\ | \n" +
52         " /  | \n" +
53         " | | \n" +
54         "=====";
55     int contador = 1, intentosTotales = 0; //USO UNA VARIABLE PARA EL CONTROL DE VIDAS Y OTRO PARA CONTAR LOS INTENTOS DENTRO DEL JUEGO
56     String letra;
57     String[] figuras = { ahor1, ahor2, ahor3, ahor4, ahor5, ahor6, ahor7 }, palabras = { "programacion", "java", "indentacion", "clases",
58     "objetos", "desarrollador", "pruebas" };
59     String palSecreta = getPalabraSecreta(palabras);
60     String[] palMatriz = new String[palSecreta.length()]; // EL USO DE UNA MATRIZ ME FACILITA LA ACTUALIZACIÓN DE LA PALABRA
61     for(int i = 0; i < palMatriz.length; i++)
62         palMatriz[i] = " ";
63     System.out.println(figuras[0]);
64     mostrarBlancos(palSecreta, palMatriz);
65     System.out.println("\n");
66     while (contador <= 6) { // MANTIENE EN BUCLE EL JUEGO HASTA QUE LA FIGURA 6 APAREZCA Y TERMINE EL JUEGO
67         letra = ingreseLetra();
68         if (letraEnPalabraSecreta(letra, palSecreta)){
69             intentosTotales++;
70             mostrarBlancosActualizados(letra, palSecreta, palMatriz);
71             if(comprobarMatrizLlena(palMatriz) == true){ //SI RETORNA QUE SE COMPLETÓ LA PALABRA, TERMINA EL JEGO
72                 System.out.println("\nADIVINASTE LA PALABRA CON " + intentosTotales + " INTENTOS. ¡FELICIDADES!");
73                 break;
74             }
75         }
76     }
77     else{
78         System.out.println(figuras[contador]);
79         contador++; //EL CONTADOR DE INTENTOS PERMITIDO AVANZARÁ Y SI SE PASA SE TERMINA EL JUEGO CON LA ÚLTIMA IMAGEN
80         intentosTotales++;
81         if(contador == 7)
82             System.out.println("\nPERDISTE! La palabra era: " + palSecreta);
83     }
84 }
85 public static String getPalabraSecreta(String[] lasPalabras) { //RETORNA UNA PALABRA SECRETA PARA EMPEZAR EL JUEGO
86     int ind, indiceMayor = lasPalabras.length-1, indiceMenor = 0;
87     ind = (int)((Math.random()*(indiceMayor-indiceMenor+1)+indiceMenor));
88     return lasPalabras[ind];
89 }
90 public static void mostrarBlancos(String palabra, String[] palMatriz) { // MUESTRA EL AVANCE DEL JUEGO
91     for (int i = 0; i < palabra.length(); i++)
92         System.out.print(palMatriz[i]);
93 }
94 public static String ingreseLetra() { // PIDE AL USUARIO INGRESAR UN DATO PARA EL JUEGO
95     String laLetra;
96     Scanner sc = new Scanner(System.in);
97     System.out.println("\nIngrese letra: ");
98     laLetra = sc.next();
99     while (laLetra.length() != 1 || Character.isDigit(laLetra.charAt(0))){ // EL BUCLE CONTINUA HASTA QUE SE INGRESE SOLO UNA LETRA
100         System.out.println("Dato erróneo, pruebe con 1 letra: ");
101         laLetra = sc.next();
102     }
103     return laLetra;
104 }
105 public static boolean letraEnPalabraSecreta(String letra, String palSecreta){ // VERIFICA SI LA LETRA INGRESADA ESTÁ DENTRO DE LA PALABRA
106     if(palSecreta.contains(letra))
107         return true;
108     return false;
109 }
110 public static void mostrarBlancosActualizados(String letra, String palSecreta, String[] palMatriz){ // CON AYUDA DE LA MATRIZ ACTUALIZA LOS BLANCOS
111     int i = palSecreta.indexOf(letra);
112     while(i < palSecreta.length()){ // ES UN BUCLE PARA ACTUALIZAR TODOS LOS ESPACIOS DONDE ESTÁ LA LETRA
113         palMatriz[i] = letra;
114         i = palSecreta.indexOf(letra, i+1);
115         if(i == -1)
116             break;
117     }
118     mostrarBlancos(palSecreta, palMatriz);
119 }
120 public static boolean comprobarMatrizLlena(String[] palMatriz){ // SE COMPROBEA SI SE COMPLETÓ LA PALABRA PARA TERMINAR EL JUEGO
121     for(int i = 0; i < palMatriz.length; i++){
122         if(palMatriz[i].equals("_ ")){
123             return false;
124         }
125     }
126     return true;
127 }

```


Dato erróneo, pruebe con 1 letra:

Ejemplo completo:

```
\redhat.java\jdt_ws\ENTREGABLE_5_7eae9+d7\bin' 'E1'
```

```
+---+  
|   |  
    |  
    |  
    |  
=====
```

```
-- -- --
```

Ingrese letra:

```
j  
j _ _ _
```

Ingrese letra:

```
a  
ja _ _
```

Ingrese letra:

```
v  
java
```



ADIVINASTE LA PALABRA CON 3 INTENTOS. ¡FELICIDADES!

D:\TAREAS FP2 JAVA\ENTREGABLES\ENTREGABLE 5>

En este largo problema, me convengo de que los arreglos son tan vitales para muchos programas, guardan un excelente orden y para un programador es imprescindible manejarlos de la mejor manera.

Mi metodología consiste, como es un programa que tiene un gran número de líneas, separar cada cosa que me pide, iniciar por requerimientos independientes y después los que dependen de este. Uso la opción de comentar bloques de código para verificar que la función por si solo funciona o me ayudo al imprimir en la terminal datos que me generarían errores para verificar que todo está corriendo bien.



REFERENCIAS Y BIBLIOGRAFÍA

	<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: 6</p>

[1]M. W. Aedo Lopez, *Fundamentos de programación, 1st ed., vol. 1. Calle Paucarpata, Puerta 5, Área de Ingenierías: EDITORIAL UNSA, 2019, p. 100.*

RUBRICA DE CALIFICACIÓN DE LABORATORIO

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. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X		
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	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	4	X	2	

	<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: 7</p>

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