
	<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:	<i>Laboratorio 2</i>				
NÚMERO DE PRÁCTICA:	<i>2</i>	AÑO LECTIVO:	<i>2024</i>	NRO. SEMESTRE:	<i>2</i>
FECHA DE PRESENTACIÓN	<i>28/09/2024</i>	HORA DE PRESENTACIÓN	<i>21/00/00</i>		
INTEGRANTE (s) <i>Karla Miluska Bedregal Coaguila</i>				NOTA (0-20)	
DOCENTE(s): <i>Lino Jose Pinto Oppe</i>					

RESULTADOS Y PRUEBAS
<p>I. EJERCICIOS RESUELTOS:</p> <p>1. JUEGO DEL AHORCADO</p> <p>En este ejercicio se le solicita a usted implementar el juego del ahorcado utilizando el código parcial que se le entrega. Deberá considerar que:</p> <ul style="list-style-type: none"> • 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 • El juego supone que el usuario no ingresa una letra ingresada previamente • El método <code>ingreseLetra()</code> debe ser modificado para incluir las consideraciones de validación • Puede crear métodos adicionales <p>Link del repositorio: https://github.com/KarlaBedregal/Bedregal_Karla_Laboratorio_02.git</p>

```

1  /* Laboratorio Nro 02 - Ejercicio 01
2  Autor: Karla Bedregal Coaguila */
3  package bedregal_karla_lab2;
4  // Version del ahorcado al 70%
5  import java.util.Scanner;
6  public class Ejercicio01 {
7      public static void main(String []args){
8          //Figuras del ahorcado
9          String ahor1 = " +---+ \n"+
10             " | | \n"+
11             " | \n"+
12             " | \n"+
13             " | \n"+
14             " | \n"+
15             "===== ";
16
17          String ahor2 = " +---+ \n"+
18             " | | \n"+
19             " O | \n"+
20             " | \n"+
21             " | \n"+
22             "===== ";
23
24          String ahor3 = " +---+ \n"+
25             " | | \n"+
26             " O | \n"+
27             " | | \n"+
28             " | | \n"+
29             " | | \n"+
30             "===== ";
31
32          String ahor4 = " +---+ \n"+
33             " | | \n"+
34             " O | \n"+
35             " /| | \n"+
36             " | | \n"+
37             " | | \n"+
38             "===== ";
39
40          String ahor5 = " +---+ \n"+
41             " | | \n"+
42             " O | \n"+
43             " /|\| \n"+
44             " | | \n"+
45             " | | \n"+
46             "===== ";
47
48          String ahor6 = " +---+ \n"+
49             " | | \n"+
50             " O | \n"+
51             " /|\| \n"+
52             " / | \n"+
53             " | | \n"+
54             "===== ";
55
56          String ahor7 = " +---+ \n"+
57             " | | \n"+
58             " O | \n"+
59             " /|\| \n"+
60             " / \| \n"+
61             " | | \n"+
62             "===== ";

```

```
63 //Arreglo de figuras
64 String [] figuras = {ahor1, ahor2, ahor3, ahor4, ahor5, ahor6, ahor7};
65 int contador = 0; //Inicializar en 0 para comenzar el juego
66 String letra;
67 //Palabras clave del juego
68 String [] palabras = {"programacion", "java", "indentacion", "clases",
69 "objetos", "desarrollador", "pruebas"};
70
71 //Obtener una palabra aleatoria
72 String palSecreta = getPalabraSecreta(palabras);
73
74 //Arreglo que almacena las letras correctas que se ingresaron
75 char[] letrasCorrectas = new char[palSecreta.length()];
76 for (int i = 0; i < letrasCorrectas.length; i++) {
77     letrasCorrectas[i] = '_'; //Representar en líneas las letras no adivinadas
78 }
79 // Bucle principal del juego
80 while(contador < 6){
81     System.out.println(figuras[contador]); //Imprimir las imagenes del ahorcado
82     mostrarBlancos(letrasCorrectas); //Mostrar espacios de la palabra
83     System.out.println("\n");
84     letra = ingreseLetra(); //Pedir al usuario que adivine una letra
85     if (letraEnPalabraSecreta(letra, palSecreta)){
86         mostrarBlancosActualizados(letra, palSecreta, letrasCorrectas);
87         //Verificar si el arreglo de letras correctas es igual a la palabra secreta
88         if (new String(letrasCorrectas).equals(palSecreta)) {
89             System.out.println("GANO - La palabra es " + palSecreta); //Mensaje de que ganó
90             return; //Salir del juego
91         }
92     } else {
93         contador = contador + 1; //Sumar los intentos fallidos
94     }
95 }
96 System.out.println(figuras[contador]);
97
```

```

97      System.out.println(figuras[contador]);
98      System.out.println("PERDIO - La palabra era " + palSecreta); //Mensaje de que perdió
99  }
100
101  //Método para obtener una palabra aleatoria del juego
102  public static String getPalabraSecreta(String [] lasPalabras){
103      int ind; //Almacenaremos el índice de la palabra aleatoria
104      int indiceMayor = lasPalabras.length - 1; //Ultimo índice del arreglo
105      int indiceMenor = 0; //Primer índice
106      ind = (int) (Math.random() * (indiceMayor - indiceMenor + 1) + indiceMenor);
107      return lasPalabras[ind]; //REtornar la palabra random
108  }
109
110  //Método para mostrar las líneas blancas del juego
111  public static void mostrarBlancos(char [] letrasCorrectas){
112      for(int i = 0; i< letrasCorrectas.length; i++) {
113          System.out.print(letrasCorrectas[i] + " "); //Mostrar las letras o espacios
114      }
115  }
116
117  //Método para ingresar y verificar si es una letra
118  public static String ingreseLetra(){
119      String laLetra;
120      Scanner sc = new Scanner(System.in);
121      System.out.println("Ingrese letra: ");
122      laLetra = sc.next();
123      //Verificar si es una sola letra
124      while(laLetra.length() != 1 || !Character.isLetter(laLetra.charAt(0))){ //Permitidos a - z
125          System.out.println("ERROR - Ingrese una letra válida"); //Mensaje de error
126          laLetra = sc.next(); //Se pide una nueva letra
127      }
128      return laLetra; //Retornar la letra que se ingresó
129  }
130

```

II. PRUEBAS

Ejecución 1:

```
Ingrese letra:
a
PROCESANDO.....
```

Ingrese letra:

Ingreso letra:

$\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$

_____ a _____

Ingrese letra:

A diagram of a vertical pipe. At the bottom, there is a horizontal line representing a piston, with the label 'piston' below it. Above the piston, there is a vertical dashed line representing the gas column. At the top of the pipe, there is a horizontal line with a small circle above it, representing the gas surface. The label 'gas' is placed to the right of the top part of the pipe.

Ingresa letra:

Diagram illustrating a simple structure with a horizontal base labeled 'a' and a vertical stem labeled 'm'. The stem has a circle at the top and a cross at the bottom.

Ingrese letra:

```
PROCESANDO.....
```

100



Increase Net

The diagram shows a simple neural network structure. It consists of two input nodes at the bottom, one hidden node in the middle, and one output node at the top. The nodes are represented by circles. The input nodes are connected to the hidden node, and the hidden node is connected to the output node. The connections are represented by lines. The output node is labeled 'E'.

Increase intra:

PERDIO - La pa

PERDIO - La palabra era clases
BUILD SUCCESSFUL (total time: 47 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: 6</p>

Ejecución 2:

```



run:
+---+
|   |
|   |
|   |
|   |
|   |
=====
- - - - -
Ingrese letra:
e
PROCESANDO.....
+---+
|   |
|   |
|   |
|   |
|   |
=====
- - - e - - -
Ingrese letra:
a
PROCESANDO.....
+---+
|   |
|   |
|   |
|   |
|   |
=====
- - - e _ a _

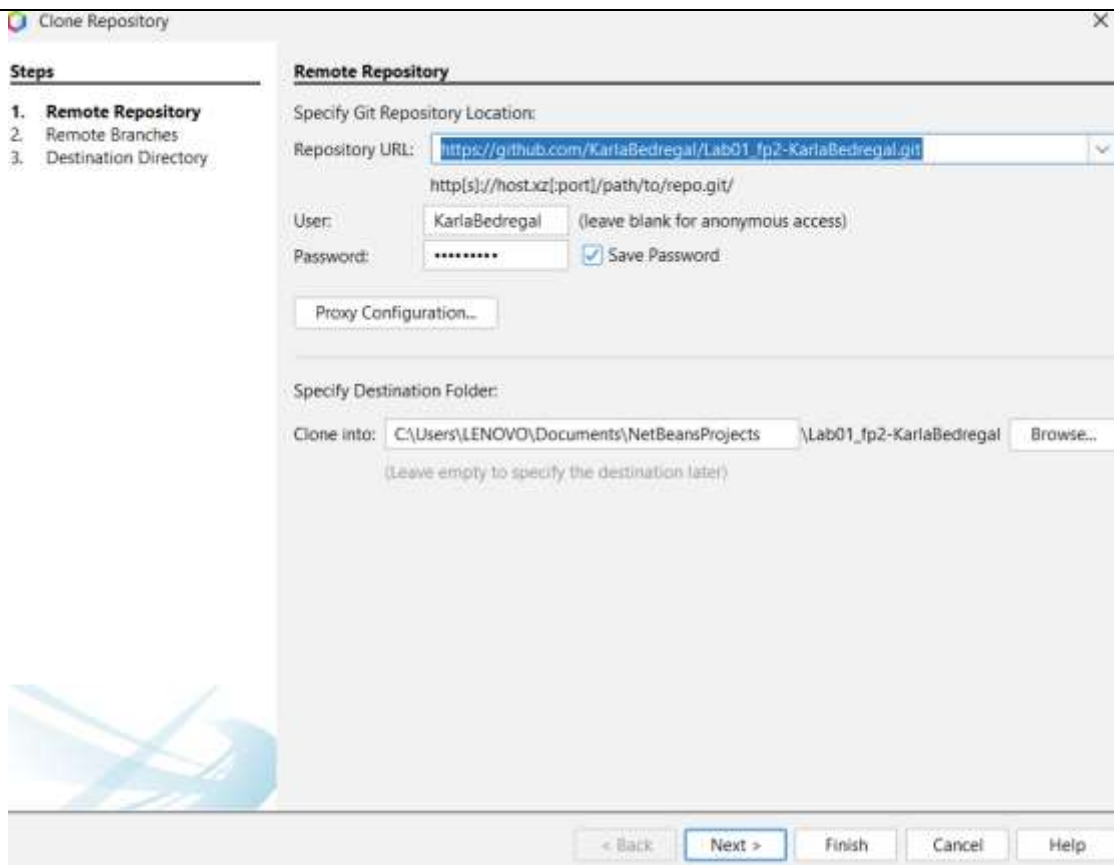
Ingrese letra:
s
PROCESANDO.....
+---+
|   |
|   |
|   |
|   |
|   |
=====
- _ _ e _ a s
Ingrese letra:
r
PROCESANDO.....
+---+
|   |
|   |
|   |
|   |
|   |
=====
_ r _ e _ a s
Ingrese letra:
p
PROCESANDO.....
+---+
|   |
|   |
|   |
|   |
|   |
=====
p r _ e _ a s

Ingrese letra:
m
+---+
|   |
|   |
|   |
|   |
|   |
=====
p r _ e _ a s
Ingrese letra:
u
PROCESANDO.....
+---+
|   |
|   |
|   |
|   |
|   |
=====
p r u e _ a s
Ingrese letra:
b
PROCESANDO.....
GANO - La palabra es pruebas
BUILD SUCCESSFUL (total time: 45 seconds)

```

Para subir el archivo a Github, en el menú superior: Team/Git/Clone, inicializamos un repositorio:

	<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>



Después de rellenar los campos, le damos a Finish, y el repositorio se subió a Github.

III. CUESTIONARIO:

1. ¿Con qué valores comprobaste que tu práctica estuviera correcta?



Probé varias veces con diferentes palabras entre las que estaban las palabras clave, para ver si funcionaba correctamente, todas eran letras, ingresadas como string pero al final cumplían con el objetivo.

2. ¿Qué resultado esperabas obtener para cada valor de entrada?

Esperaba que, en primer lugar, al iniciar el juego, aparecieran en orden las figuras, los métodos funcionaran, los blancos y las líneas también, que se contaran los intentos, y que, si al adivinar en 6 veces, el usuario perdiera; caso contrario, ganara la partida.

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

En cada valor de entrada, se verificó si el string fuera parte de la palabra clave ingresada. Si era el caso, se continuaba con el bucle hasta completar la palabra. Si no, se contabilizaban los errores hasta llegar a un límite de 6 (cuando se formaba la figura completa del ahorcado), entonces se perdía el juego.

	<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

CONCLUSIONES

Al finalizar este laboratorio, hemos explorado la funcionalidad de un juego del ahorcado haciendo uso de métodos, clases, arreglos. Al ser un juego elaborado en un 70%, solo faltaron algunos ajustes para completar y verificar si funcionara correctamente.

Además, hemos analizado nuevos métodos de la clase Character, como el isLetter(), para determinar si un carácter en específico era una letra o no. Así, 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.

RUBRICA PARA EL CONTENIDO DEL INFORME Y DEMOSTRACIÓN

El alumno debe marcar o dejar en blanco en celdas de la columna Checklist si cumplió con el ítem correspondiente.



Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.

El alumno debe autocalificarse en la columna Estudiante de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

Nivel				
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

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. (El profesor puede preguntar para refrendar calificación).	4	X	1	
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	

	<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: 10

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		16	

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