





| | | |
|--|--|---|
|  | <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

| INFORMACIÓN BÁSICA | | | | | |
|--|-----------------------------------|----------------------|----------|----------------|---|
| ASIGNATURA: | Fundamentos de la Programación 02 | | | | |
| TÍTULO DE LA PRÁCTICA: | Arreglos Estándar | | | | |
| NÚMERO DE PRÁCTICA: | 02 | AÑO LECTIVO: | 2024 | NRO. SEMESTRE: | 2 |
| FECHA DE PRESENTACIÓN | 27/09/2024 | HORA DE PRESENTACIÓN | 14/41/00 | | |
| INTEGRANTE (s) German Arturo Chipana Jerónimo | | | | NOTA (0-20) | |
| DOCENTE(s): Pinto Oppe Lino José | | | | | |

| RESULTADOS Y PRUEBAS |
|---|
| <p>I. EJERCICIOS RESUELTOS:</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>EJERCICIO 01:</p> <p>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 |

```
Start Page x Ejercicio1Lab02.java x
Source History
1  /*
2     Autor: Chipana Jeronimo German Arturo
3     Proposito: Ejercicio 1 (JUEGO DEL AHORCADO)
4  */
5  package ejerciciolab02;
6
7  import java.util.*;
8  public class Ejercicio1Lab02 {
9
10     public static void main(String[] args) {
11         // Definición de las diferentes etapas del dibujo del ahorcado
12         String ahor1 = """
13             +---+
14             | |
15             |
16             |
17             |
18             |
19             ===== """;
20         String ahor2 = """
21             +---+
22             | |
23             O |
24             |
25             |
26             |
27             ===== """;
28         String ahor3 = """
29             +---+
30             | |
31             O |
32             | |
33             |
34             |
35             ===== """;
36         String ahor4 = """
37             +---+
38             | |
39             O |
40             /| |
41             |
42             |
43             ===== """;
44         String ahor5 = """
45             +---+
46             | |
47             O |
48             /|\ |
49             |
50             |
51             ===== """;
52         String ahor6 = """
53             +---+
54             | |
55             O |
56             /|\ |
```

| | | |
|---|---|---|
|  | <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 style="text-align: center;">Aprobación: 2022/03/01</p> | <p style="text-align: center;">Código: GUIA-PRLE-001</p> | <p style="text-align: right;">Página: 3</p> |

Start Page x Ejercicio1lab02.java x

Source History

Projects - LABORATORIO 2

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

```

55         o |
56         /|\ |
57         / |
58         |
59         =====
60 String ahor7 = ""
61         +---+
62         | |
63         o |
64         /|\ |
65         / \ |
66         |
67         =====
68 String [] figuras = {ahor1, ahor2, ahor3, ahor4, ahor5, ahor6, ahor7}; // Arreglo que contiene todas las figuras del ahorcado.
69 int contador = 0; // Contador de errores (número de intentos fallidos)
70 String letra; // Variable para almacenar la letra ingresada por el usuario.
71 // Arreglo de palabras posibles para el juego.
72 String [] palabras = {"programacion", "java", "indentacion", "clases",
73 "objetos", "desarrollador", "pruebas"};
74 String palSecreta = getPalabraSecreta(palabras); // Selección aleatoria de la palabra secreta.
75 boolean[] aciertos = new boolean[palSecreta.length()]; // Arreglo booleano para rastrear las letras acertadas.
76 System.out.println(figuras[0]); // Muestra la primera figura del ahorcado.
77 mostrarBlancos(palSecreta,aciertos); // Muestra la palabra secreta con letras ocultas.
78 System.out.println("\n");
79 // Bucle principal del juego que continúa hasta que se alcance el límite de errores
80 // o se adivine la palabra completa.
81 while(contador < 6 && !juegoTerminado(aciertos)){
82     letra = ingreseLetra();
83     if (letraEnPalabraSecreta(letra, palSecreta))
84         // Si la letra está en la palabra secreta, actualiza los aciertos.
85         actualizarBlancos(letra,palSecreta,aciertos);
86     else{
87         // Si la letra no está, incrementa el contador de errores y muestra la siguiente figura.
88         contador = contador +1;
89         System.out.println(figuras[contador]);
90     }
91     // Muestra la palabra con las letras acertadas hasta el momento.
92     mostrarBlancos(palSecreta, aciertos);
93     System.out.println("\n");
94 }
95 System.out.println("\n");
96 // Verifica si el juego terminó por adivinar la palabra o por agotar los intentos.
97 if (juegoTerminado(aciertos)) {
98     System.out.println("Felicitades! Has adivinado la palabra.");
99 }
100 else {
101     System.out.println("Lo siento, has perdido. La palabra era: " + palSecreta);
102 }
103 }
```

```

103     }
104     // Método para seleccionar aleatoriamente una palabra secreta del arreglo proporcionado.
105     public static String getPalabraSecreta(String [] lasPalabras){
106         String palSecreta;
107         int ind;
108         int indiceMayor = lasPalabras.length -1;
109         int indiceMenor =0;
110         ind = (int) ((Math.random() * (indiceMayor - indiceMenor + 1) + indiceMenor));
111         return lasPalabras[ind]; // Retorna una palabra seleccionada aleatoriamente.
112     }
113     // Método para mostrar la palabra secreta con letras acertadas y espacios en blanco.
114     public static void mostrarBlancos(String palabra, boolean[] aciertos){
115         for (int i = 0; i < palabra.length(); i++) {
116             if (aciertos[i]) {
117                 System.out.print(palabra.charAt(i) + " ");
118             } else {
119                 System.out.print("_ ");
120             }
121         }
122     }
123     // Método para solicitar al usuario que ingrese una letra válida.
124     public static String ingreseLetra(){
125         String laLetra;
126         Scanner sc = new Scanner(System.in);
127         System.out.println("Ingrese letra: ");
128         laLetra = sc.next();
129         //Valida que el usuario ingrese solo una letra y que sea una letra del alfabeto.
130         while(laLetra.length() != 1 || !Character.isLetter(laLetra.charAt(0))){
131             System.out.println("Ingrese letra valida: ");
132             laLetra = sc.next();
133         }
134         return laLetra; //Retorna la letra ingresada por el usuario.
135     }
136     // Método para verificar si una letra está presente en la palabra secreta.
137     public static boolean letraEnPalabraSecreta(String letra, String palSecreta ) {
138         for(int i=0;i<palSecreta.length();i++){
139             if(palSecreta.charAt(i)==letra.charAt(0))
140                 return true; // Retorna true si la letra se encuentra en la palabra.
141         }
142         return false; // Retorna false si la letra no se encuentra.
143     }
144     // Método para actualizar el arreglo de aciertos con la letra acertada.
145     public static void actualizarBlancos(String letra,String palSecreta, boolean[] aciertos){
146         System.out.println("PROCESANDO.....");
147         for(int i=0; i< palSecreta.length(); i++){
148             if(letra.charAt(0)==palSecreta.charAt(i)){
149                 aciertos[i]=true; // Marca la posición como acertada.
150             }
151         }
152     }
153     // Método para verificar si el juego ha terminado, es decir, si todas las letras han sido acertadas.
154     public static boolean juegoTerminado(boolean[] aciertos) {
155         for (boolean acierto : aciertos) {
156             if (!acierto) {
157                 return false; // Retorna false si hay al menos una letra sin acertar.
158             }
159         }
160         return true; // Retorna true si todas las letras han sido acertadas.
161     }
162 }



```

II. PRUEBAS

¿Con que valores comprobaste que tu práctica estuviera correcta?

Comprobé ingresando cadenas de más de una palabra y cadenas de números.

¿Qué resultado esperabas obtener para cada valor de entrada?

| | | |
|---|---|---|
|  | <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: 5</p> |

Esperaba que cuando se ingrese una letra no valida, el juego me la vuelva a pedir hasta ser una letra valida. También que cuando la letra no este en la palabra secreta, que sume una figura del ahorcado hasta perder. Por último que las letras correctas se muestren revelando los espacios en blancos.

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

Que las letras correctas me revelen los espacios en blanco de la palabra secreta, y que las letras que no estaban en la palabra secreta sumen una figura del ahorcado.

Por último, que al finalizar el juego se muestre si el usuario pudo adivinar la palabra secreta o no la pudo.

```

run:
+---+
| |
| |
| |
| |
| |
=====
- - - -

Ingrese letra:
a
PROCESANDO.....
_ a _ a

Ingrese letra:
7
Ingrese letra valida:
por
Ingrese letra valida:
j
PROCESANDO.....
j a _ a

Ingrese letra:
y
+---+
| |
O |
| |
| |
| |
=====
j a _ a

Ingrese letra:
v
PROCESANDO.....
j a v a

Felicidades! Has adivinado la palabra.
BUILD SUCCESSFUL (total time: 24 seconds)

```

Es esta prueba ingresamos valores de numero o cadenas de mas de una letra, en las cuales nos dice que volvamos a introducir una letra valida.

Projects - LABORAT

Output - Ejercicio1Lab02 (run) ×

```

run:
+----+
| |
| |
| |
| |
| |
=====
-----
Ingrese letra:
l
Ingrese letra valida:
t
PROCESANDO.....
----- t -----

Ingrese letra:
a
PROCESANDO.....
----- t a -----

Ingrese letra:
p
+----+
| |
| |
| |
| |
| |
=====
----- t a -----

Ingrese letra:
o
PROCESANDO.....
----- t a _ o _

Ingrese letra:
c
PROCESANDO.....
----- t a c _ o _

Ingrese letra:
i
PROCESANDO.....
i _ _ _ t a c i o _

Ingrese letra:
n
PROCESANDO.....
i n _ _ n t a c i o n



Ingrese letra:
d
PROCESANDO.....
i n d _ n t a c i o n

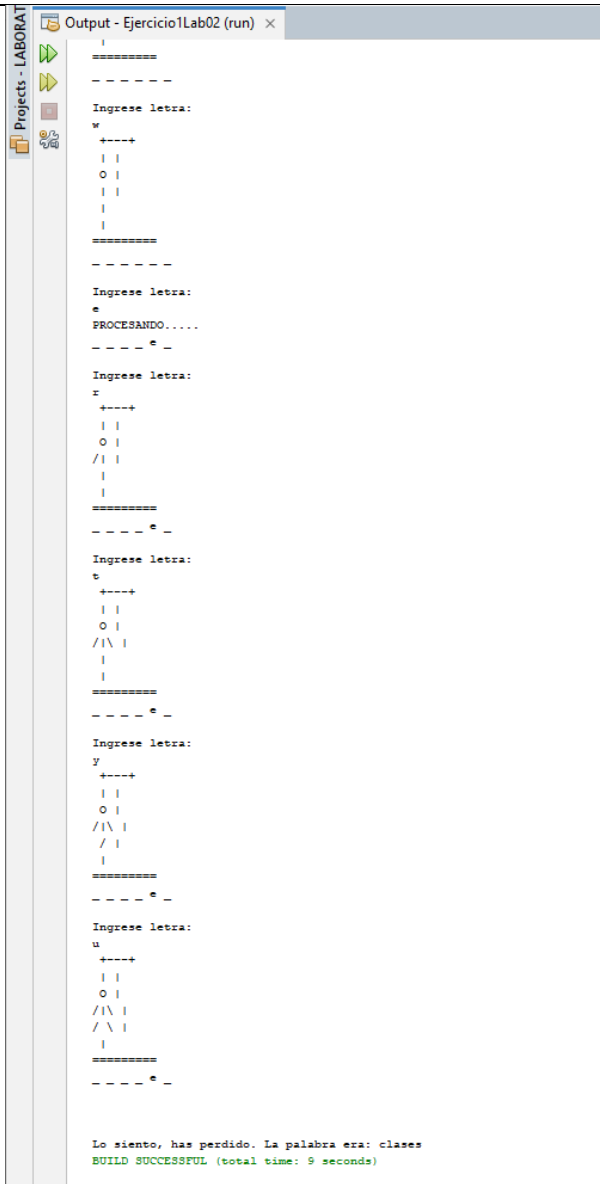
Ingrese letra:
e
PROCESANDO.....
i n d e n t a c i o n

Felicitades! Has adivinado la palabra.
BUILD SUCCESSFUL (total time: 36 seconds)

```



En esta prueba podemos notar como se van llenando los espacios en blanco de la palabra secreta, y finalmente nos manda un mensaje diciendo que adivinamos la palabra secreta.

| | | |
|--|--|---|
|  | <p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p> |  |
| <p align="center">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p> | | |
| <p>Aprobación: 2022/03/01</p> | <p align="center">Código: GUIA-PRLE-001</p> | <p align="right">Página: 7</p> |



En esta prueba podemos notar como el usuario ingresa letras validas pero que no están dentro de la palabra secreta, hasta llenar las figuras. Finalmente nos manda un mensaje diciendo que perdimos y cual era la palabra secreta.

Mi commit:

| | | |
|---|--|---|
|  | <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: 8</p> |

```

user@DESKTOP-CS7PIHP MINGW64 ~
$ cd Desktop

user@DESKTOP-CS7PIHP MINGW64 ~/Desktop
$ cd CHIPANA_JERONIMO_LABORATORIO_02_FP2

user@DESKTOP-CS7PIHP MINGW64 ~/Desktop/CHIPANA_JERONIMO_LABORATORIO_02_FP2
$ git init
Initialized empty Git repository in C:/Users/user/Desktop/CHIPANA_JERONIMO_LABORATORIO_02_FP2/.git/

user@DESKTOP-CS7PIHP MINGW64 ~/Desktop/CHIPANA_JERONIMO_LABORATORIO_02_FP2 (master)
$ git commit -m "Juego del Ahorcado Laboratorio 02"
[master (root-commit) a49e2a9] Juego del Ahorcado Laboratorio 02
1 file changed, 164 insertions(+)
create mode 100644 Ejercicio1Lab02.java

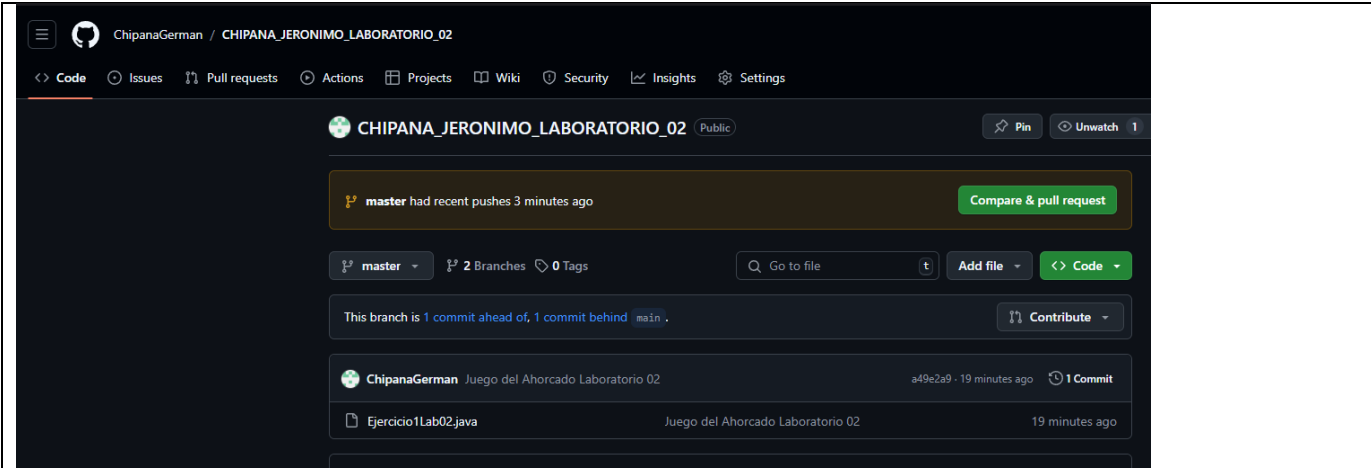
user@DESKTOP-CS7PIHP MINGW64 ~/Desktop/CHIPANA_JERONIMO_LABORATORIO_02_FP2 (master)
$ git remote set-url origin https://github.com/ChipanaGerman/CHIPANA_JERONIMO_LABORATORIO_02.git

user@DESKTOP-CS7PIHP MINGW64 ~/Desktop/CHIPANA_JERONIMO_LABORATORIO_02_FP2 (master)
$ git remote -v
origin https://github.com/ChipanaGerman/CHIPANA_JERONIMO_LABORATORIO_02.git (fetch)
origin https://github.com/ChipanaGerman/CHIPANA_JERONIMO_LABORATORIO_02.git (push)

user@DESKTOP-CS7PIHP MINGW64 ~/Desktop/CHIPANA_JERONIMO_LABORATORIO_02_FP2 (master)
$ git status
On branch master
nothing to commit, working tree clean

user@DESKTOP-CS7PIHP MINGW64 ~/Desktop/CHIPANA_JERONIMO_LABORATORIO_02_FP2 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 2.00 KiB | 2.00 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote: https://github.com/ChipanaGerman/CHIPANA_JERONIMO_LABORATORIO_02/pull/new/master
remote:
To https://github.com/ChipanaGerman/CHIPANA_JERONIMO_LABORATORIO_02.git
 * [new branch]      master -> master

```

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 | 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 | 1 | |
| 2. Commits | Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. | 4 | X | 3 | |

| | | | | | |
|------------------|--|----|---|----|--|
| | (El profesor puede preguntar para refrendar calificación). | | | | |
| 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 acabado impecable. (El profesor puede preguntar para refrendar calificación). | 4 | X | 4 | |
| TOTAL | | 20 | | 18 | |

CONCLUSIONES

Colocar las conclusiones, apreciaciones reflexivas, opiniones finales a cerca de los resultados obtenidos de la sesión de laboratorio.

CONCLUSIÓN:

| | | |
|--|--|---|
|  | <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: 11 |

La implementación del juego del ahorcado permitió aplicar conceptos como arreglos, ciclos y validación de entradas. El juego funciona correctamente, mostrando las letras acertadas y los errores del usuario mediante la figura del ahorcado. Esta práctica refuerza el entendimiento de la lógica de control y manejo de datos, demostrando un buen dominio en la solución de problemas.

METODOLOGÍA DE TRABAJO

Colocar la metodología de trabajo que ha utilizado el estudiante o el grupo para resolver la práctica, es decir el procedimiento/secuencia de pasos en forma general.

- Se analizó el problema y se definió la estructura del programa, identificando los métodos necesarios, como la validación de letras, actualización de aciertos y gestión de errores.
- Se implementó el código en Java, utilizando arreglos para gestionar las palabras secretas y booleanos para rastrear los aciertos del usuario.
- Se realizaron múltiples pruebas ingresando letras válidas e inválidas, así como simulando escenarios donde el usuario ganaba o perdía, verificando que el juego respondiera correctamente en cada caso.
- Finalmente, se corrigieron errores y se documentó el código.

REFERENCIAS Y BIBLIOGRAFÍA

Colocare las referencias utilizadas para el desarrollo de la práctica en formato IEEE

M. Aedo López, Práctica de Laboratorio 2: Arreglos Estándar, Universidad Nacional de San Agustín, 2023.