




```
1 package ejercicios;
2 //Laboratorio N° 02 -Ejercicio1
3 //Autor: Jose Manuel Morocco Saico
4 import java.util.Scanner;
5 public class ejercicio1 {
6     public static void main(String[] args) {
7
8         //Etapas el ahorcado:
9         String ahor1 = " +---+ \n" +
10             " |   | \n" +
11             " |   | \n" +
12             " |   | \n" +
13             " |   | \n" +
14             " |   | \n" +
15             "===== ";
16         String ahor2 = " +---+ \n" +
17             " |   | \n" +
18             " 0   | \n" +
19             " |   | \n" +
20             " |   | \n" +
21             " |   | \n" +
22             "===== ";
23
24         String ahor3 = " +---+ \n" +
25             " |   | \n" +
26             " 0   | \n" +
27             " |   | \n" +
28             " |   | \n" +
29             " |   | \n" +
30             "===== ";
31
32         String ahor4 = " +---+ \n" +
33             " |   | \n" +
34             " 0   | \n" +
35             " /|  | \n" +
36             " |   | \n" +
37             " |   | \n" +
38             "===== ";
39
40         String ahor5 = " +---+ \n" +
41             " |   | \n" +
42             " 0   | \n" +
43             " /|\ \n" +
44             " |   | \n" +
45             " |   | \n" +
46             "===== ";
```

	<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: 3</p>

```

48     String ahor6 = " +---+ \n" +
49         " |   | \n" +
50         " 0   | \n" +
51         "/|\\  | \n" +
52         "/   | \n" +
53         "    | \n" +
54         "=====";
55
56     String ahor7 = " +---+ \n" +
57         " |   | \n" +
58         " 0   | \n" +
59         "/|\\  | \n" +
60         "/ \\  | \n" +
61         "    | \n" +
62         "=====";
63
64     // Arreglo con todas las figuras del ahorcado
65     String[] figuras = { ahor1, ahor2, ahor3, ahor4, ahor5, ahor6, ahor7 };
66
67     int contador = 0; // Contador de errores
68     String letra; // Variable para la letra ingresada por el usuario
69
70     // Lista de palabras que pueden ser seleccionadas para el juego
71     String[] palabras = { "programacion", "java", "indentacion", "clases",
72         "objetos", "desarrollador", "pruebas" };
73
74     // Seleccionar una palabra secreta al azar
75     String palSecreta = getPalabraSecreta(palabras);
76     System.out.println(figuras[0]);
77
78     // Inicializar la palabra oculta con guiones bajos
79     String palabraPorAdivinar=generarBlancos(palSecreta.length());
80     System.out.println(palabraPorAdivinar);
81     System.out.println("\n");
82
83



```

```

83 // Bucle principal del juego: se repite mientras no se supere el límite de
84 //errores y no se haya adivinado la palabra
85 while (contador < 6 && palabraPorAdivinar.contains("_")) {
86     letra = ingreseLetra();
87
88     // Verificar si la letra está en la palabra secreta
89     if (LetraEnPalabraSecreta(letra, palSecreta)) {
90         // Actualizar la palabra por adivinar con la letra encontrada
91         palabraPorAdivinar = actualizarBlancos(letra, palabraPorAdivinar, palSecreta);
92     } else {
93         contador++; // Incrementar el contador de errores
94     }
95
96     // Mostrar la figura y el progreso después de cada intento
97     System.out.println(figuras[contador]); // Mostrar la figura correcta
98     System.out.println(palabraPorAdivinar); // Mostrar la palabra con guiones y letras adivinadas
99 }
100
101 // Verificar si perdió o ganó
102 if (palabraPorAdivinar.contains("_")) {
103     System.out.println("Perdiste. La palabra secreta era: " + palSecreta);
104 } else {
105     System.out.println("¡Ganaste! Adivinaste la palabra en " + (contador + 1) + " turnos.");
106 }
107 System.out.println("\n");
108 }
109
110 // Método que selecciona una palabra secreta al azar del arreglo de palabras
111 public static String getPalabraSecreta(String[] lasPalabras) {
112     int ind;
113     int indiceMayor = lasPalabras.length - 1;
114     int indiceMenor = 0;
115     ind = (int) ((Math.random() * (indiceMayor - indiceMenor + 1)) + indiceMenor);
116     return lasPalabras[ind];
117 }
118
119 // Método que genera una cadena de guiones bajos del mismo tamaño que la palabra secreta
120 public static String generarBlancos(int longitud) {
121     String blancos = "";
122     String blanco = "_";
123     for (int i = 0; i < longitud; i++) {
124         blancos += blanco;
125     }
126     return blancos;
127 }

```

```
130 // Método que solicita una letra válida al usuario
131 public static String ingreseLetra() {
132     String laLetra;
133     Scanner sc = new Scanner(System.in);
134     System.out.println("Ingresa letra: ");
135     laLetra = sc.next();
136     char letra=laLetra.charAt(0);
137
138     // Asegurarse de que la entrada sea una única letra y no un número
139     while (laLetra.length() != 1 || Character.isDigit(letra)) {
140         System.out.println("Ingresa letra: ");
141         laLetra = sc.next();
142         letra=laLetra.charAt(0);
143     }
144     return laLetra;
145 }
146
147 // Método que verifica si la letra está en la palabra secreta
148 public static boolean letraEnPalabraSecreta(String letra, String palSecreta) {
149     for(int i=0;i<palSecreta.length();i++) {
150         if(letra.charAt(0)==palSecreta.charAt(i))
151             return true;
152     }
153     return false;
154 }
155
156 // Método que actualiza los guiones bajos de la palabra por adivinar con la letra correcta
157 public static String actualizarBlancos(String letra,String palabraPorAdivinar, String palSecreta) {
158     char[] blancosActualizados=palabraPorAdivinar.toCharArray();// Convertir palabra a adivinar en array de caracteres
159     for(int i=0;i<palSecreta.length();i++) {
160         if (palSecreta.charAt(i) == letra.charAt(0)) {
161             blancosActualizados[i]=letra.charAt(0);// Actualizar las posiciones correctas con la letra ingresada
162         }
163     }
164     return new String(blancosActualizados);// Devolver la palabra actualizada como cadena
165 }
166 }
167 }
```

	<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>

II. PRUEBAS

En caso de adivinar la palabra:

```

+---+
|   |
+---+
=====
o___o
Ingrese letra:
o
+---+
|   |
+---+
=====
o___o
Ingrese letra:
b
+---+
|   |
+---+
=====
ob___o
Ingrese letra:
i
+---+
|   |
+---+
=====
ob___o
Ingrese letra:
e
+---+
|   |
+---+
=====
ob_e_o_
Ingrese letra:
j
+---+
|   |
+---+
=====
obje_o_
Ingrese letra:
s
+---+
|   |
+---+
=====
obje_os
Ingrese letra:
t
+---+
|   |
+---+
=====
objetos
¡Ganaste! Adivinaste la palabra en 2 turnos.

```

```

+---+
|   |
+---+

=====

Ingrese letra:
x
+---+
|   |
0   |
+---+

=====

Ingrese letra:
x
+---+
|   |
0   |
|   |
+---+

=====

Ingrese letra:
x
+---+
|   |
0   |
/|  |
+---+

=====

Ingrese letra:
x
+---+
|   |
0   |
/|\ |
+---+

=====

Ingrese letra:
x
+---+
|   |
0   |
/|\ |
/  |
+---+

=====

Ingrese letra:
x
+---+
|   |
0   |
/|\ |
/  |
+---+



=====

```

Verificación al introducir caracteres

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

Ingrese letra:
6
Ingrese letra:
a
+---+
|
|
|
=====
_a_a
Ingrese letra:
ja
Ingrese letra:
j
+---+
|
|
|
=====
ja_a
Ingrese letra:
v
+---+
|
|
|
=====
java
;Ganaste! Adivinaste la palabra en 1 turnos.
```


	<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: 9</p>

Mi commit al repositorio :



```
Windows@DESKTOP-C32KACO MINGW64 ~/Desktop/nuevoclipse/FP2_LAB02_JOSEMOROCCO (master)
$ git branch -M main

Windows@DESKTOP-C32KACO MINGW64 ~/Desktop/nuevoclipse/FP2_LAB02_JOSEMOROCCO (main)
$ git remote add origin https://github.com/JoseMorocco/FP2.git

Windows@DESKTOP-C32KACO MINGW64 ~/Desktop/nuevoclipse/FP2_LAB02_JOSEMOROCCO (main)
$ git push -u origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (15/15), 5.37 KiB | 2.68 MiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/JoseMorocco/FP2.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

III. RUBRICA:

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. (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	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	

	<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: 11</p>

CONCLUSIONES

Al resolver el programa pude darme cuenta que es muy importante mantener el orden a la hora de elaborarlo pues en varios momentos tuve complicaciones con los métodos que debía usar, además de la gran importancia de los arreglos

METODOLOGÍA DE TRABAJO

- 1.-Evaluar las partes incompletas o a mejorar del código
- 2.-Elaborar un pseudocódigo
- 3.-Elaborar un diagrama de Flujo sobre como debería ir el juego
- 4.-Aplicar y corregir errores

REFERENCIAS Y BIBLIOGRAFÍA