



```
package laboratorio_2;
//Laboratorio N° 02 -Ejercicio 01
//Autor: Usiel Surriel Quispe Puma
import java.util.Scanner;

public class EJERCICIO_1 {

    public static void main(String[] args) {

        String ahor1
            = " +---+ \n"
            + " |   | \n"
            + " |   | \n"
            + " |   | \n"
            + " |   | \n"
            + " |   | \n"
            + " ===== ";

        String ahor2
            = " +---+ \n"
            + " |   | \n"
            + " 0   | \n"
            + " |   | \n"
            + " |   | \n"
            + " |   | \n"
            + " ===== ";

        String ahor3
            = " +---+ \n"
            + " |   | \n"
            + " 0   | \n"
            + " |   | \n"
            + " |   | \n"
            + " |   | \n"
            + " ===== ";

        String ahor4 = " +---+ \n"
            + " |   | \n"
            + " 0   | \n"
            + " /|  | \n"
            + " |   | \n"
            + " |   | \n"
            + " ===== ";
```

```
String ahor5
= " +---+ \n"
+ " | | \n"
+ " 0 | \n"
+ " /|\ | \n"
+ " | | \n"
+ " | \n"
+ "=====";

String ahor6
= " +---+ \n"
+ " | | \n"
+ " 0 | \n"
+ " /|\ | \n"
+ " / | \n"
+ " | \n"
+ "=====";

String ahor7
= " +---+ \n"
+ " | | \n"
+ " 0 | \n"
+ " /|\ | \n"
+ " / \ | \n"
+ " | \n"
+ "=====";

// Arreglos para las figuras del ahorcado y palabras secretas
String[] figuras = {ahor1, ahor2, ahor3, ahor4, ahor5, ahor6, ahor7};
String[] palabras = {"programacion", "java", "indentacion", "clases", "objetos", "desarrollador", "pruebas"};

String letra, palSecreta = getPalabraSecreta(palabras);
String palNueva = "";
for (int i = 0; i < palSecreta.length(); i++) {
    // Inicializa la palabra actual con guiones bajos
    palNueva += "_ ";
}

int contador = 0, turnosFallidos = 0;

System.out.println(figuras[contador]);
System.out.println(palNueva);
```

```
while (contador < figuras.length - 1) { // Ciclo principal del juego
    letra = ingreseLetra(); // Solicita al usuario una letra
    if (letraEnPalabraSecreta(letra, palSecreta)) {
        System.out.println("Encontrado");
        palNueva = mostrarBlancosActualizados(letra, palSecreta, palNueva);
    } else {
        contador++;
        // Muestra el siguiente estado del ahorcado
        System.out.println(figuras[contador]);
        turnosFallidos++;
    }

    // Muestra los espacios actualizados después de procesar la letra
    mostrarBlancos(palSecreta, palNueva);
    if (quitarEspacios(palNueva).equals(palSecreta)) { // Verifica si se ha ganado
        System.out.println("=====\\n");
        System.out.println("GANASTE! Adivinaste la palabra: " + palSecreta);
        System.out.println("Turnos Fallidos: " + turnosFallidos);
        break;
    }
}

if (contador == figuras.length - 1) { // Si se llegó al máximo de intentos
    System.out.println("=====\\n");
    System.out.println("Has perdido, La palabra era: " + palSecreta);
}

// Método para obtener una palabra secreta aleatoria
public static String getPalabraSecreta(String[] lasPalabras) {
    int ind = (int) (Math.random() * lasPalabras.length);
    return lasPalabras[ind];
}

// Método para mostrar los espacios en blanco de la palabra actual
public static void mostrarBlancos(String palabraSecreta, String palabraActual) {
    for (int i = 0; i < palabraSecreta.length(); i++) {
        if (palabraActual.charAt(i * 2) == '_' ) {
            System.out.print("_ ");
        } else {
            System.out.print(palabraActual.charAt(i * 2) + " ");
        }
    }
}
```

```
        System.out.println();
    }

    // Método para ingresar una letra del usuario
    public static String ingreseLetra() {
        Scanner sc = new Scanner(System.in);
        String laLetra;

        while (true) {
            System.out.print("\nIngrese letra: ");
            laLetra = sc.next();
            if (laLetra.length() == 1 && Character.isLetter(laLetra.charAt(0))) {
                return laLetra.toLowerCase();
            } else {
                System.out.println("Error! Solo se permite una letra");
            }
        }
    }

    // Método para verificar si la letra está en la palabra secreta
    public static boolean letraEnPalabraSecreta(String letra, String palSecreta) {
        for (int i = 0; i < palSecreta.length(); i++) {
            if (palSecreta.charAt(i) == letra.charAt(0)) {
                return true;
            }
        }
        return false;
    }

    // Método para actualizar el estado de la palabra al adivinar una letra
    public static String mostrarBlancosActualizados(String letra, String palSecreta, String palNueva) {
        String nuevoEstado = "";
        for (int i = 0; i < palSecreta.length(); i++) {
            if (palSecreta.charAt(i) == letra.charAt(0)) {
                nuevoEstado += letra + " ";
            } else {
                nuevoEstado += (palNueva.charAt(i * 2) != '_' ? palNueva.charAt(i * 2) : "_") + " ";
            }
        }
        return nuevoEstado;
    }

    // Método para quitar los espacios de la palabra actual
    public static String quitarEspacios(String palNueva) {
        String palSinEspacios = "";
        for (int i = 0; i < palNueva.length(); i++) {
            if (palNueva.charAt(i) != ' ') {
                palSinEspacios += palNueva.charAt(i);
            }
        }
        return palSinEspacios;
    }
}
```

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

-- -- --

Ingrese letra: j  
Encontrado  
j \_ \_ \_

Ingrese letra: 43  
Error! Solo se permite una letra

Ingrese letra: 3  
Error! Solo se permite una letra

Ingrese letra: a  
Encontrado  
j a \_ a

Ingrese letra: v  
Encontrado  
j a v a

=====

GANASTE ! adivinaste la palabra: java  
Turnos Fallidos : 0

```

Ingreso letra: b
+---+
|   |
|   |
0   |
/|\ |
    |
    |
=====
_ _ v _

Ingreso letra: n
+---+
|   |
|   |
0   |
/|\ |
/   |
    |
    |
=====
_ _ v _

Ingreso letra: c
+---+
|   |
|   |
0   |
/|\ |
/ \ |
    |
    |
=====
_ _ v _
=====

Has perdido, La palabra era: java

```

## II. PRUEBAS

## III. CUESTIONARIO:

## IV. RÚBRICA:

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		1	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4		2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2		3	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2		3	
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		3	
7. Ortografía	El documento no muestra errores ortográficos.	2		3	
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		20	

## CONCLUSIONES



	<p 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>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 9

En el desarrollo de este laboratorio, pude ver nuevas formas de arreglar códigos ya iniciados, lo cual considero importante al analizar códigos hechos por otras personas.

### METODOLOGÍA DE TRABAJO

La metodología que use para realizar este laboratorio fue , ver los ejercicios y sus restricciones después identificar las posibles soluciones para que el programa funcione, aunque tuve dificultades pude hacer que los códigos funcionaran.

### REFERENCIAS Y BIBLIOGRAFÍA