

	<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

(formato estudiante)

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	30/09/2024	HORA DE PRESENTACIÓN	hh/mm/ss		
INTEGRANTE (s) Palma Apaza, Santiago Enrique				NOTA (0-20)	Nota colocada por el docente
DOCENTE(s): Pinto Oppe Lino José					

RESULTADOS Y PRUEBAS
<p>I. EJERCICIOS RESUELTOS:</p> <p>El estudiante coloca la evidencia de los ejercicios propuestos realizados en la sesión de laboratorio, en el tiempo o duración indicado por el docente.</p> <p>El docente debe colocar la retroalimentación por cada ejercicio que el estudiante/grupo ha presentado</p> <p>JUEGO DEL AHORCADO En este ejercicio se le solicita a usted implementar el juego del ahorcado utilizando el código parcial que se le entrega.</p> <p>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

```

20 Autor: Santiago Palma Apaza
5 package labFP2;
6
7 import java.util.*;
8
9 public class Ahorcado {
10
11     public static void main(String[] args) {
12         String etapa1 = """
13             +---+
14             |
15             |
16             |
17             |
18             |
19             ===== """;
20         String etapa2 = """
21             +---+
22             |
23             0
24             |
25             |
26             |
27             ===== """;
28         String etapa3 = """
29             +---+
30             |
31             0
32             |
33             |
34             |
35             ===== """;
36         String etapa4 = """
37             +---+
38             |
39             0
40             /|
41             |
42             |
43             ===== """;
44         String etapa5 = """
45             +---+
46             |
47             0
48             /|\
49             |
50             |
51             ===== """;
52         String etapa6 = """
53             +---+
54             |
55             0
56             /|\

```

listaPalabras[]: Este array contiene las posibles palabras que se pueden elegir aleatoriamente para el juego

palabraOculta: Aquí se guarda la palabra secreta

letrasDescubiertas[]: Es un array de booleanos que lleva un registro de qué letras han sido adivinadas

	<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: 3</p>

```

public static String ingresarLetra() {
    Scanner teclado = new Scanner(System.in);
    System.out.println("Ingresa una letra: ");
    String letra = teclado.next();

    while (letra.length() != 1 || !Character.isLetter(letra.charAt(0))) {
        System.out.println("ingrese una letra válida: ");
        letra = teclado.next();
    }

    return letra;
}

public static boolean existeLetraEnPalabra(String letra, String palabraSecreta) {
    return palabraSecreta.contains(letra);
}

public static void actualizarLetrasDescubiertas(String letra, String palabraSecreta, boolean[] descubiertas) {
    System.out.println("Verificando...");
    for (int i = 0; i < palabraSecreta.length(); i++) {
        if (letra.charAt(0) == palabraSecreta.charAt(i)) {
            descubiertas[i] = true;
        }
    }
}

public static boolean esJuegoFinalizado(boolean[] descubiertas) {
    for (boolean letraDescubierta : descubiertas) {
        if (!letraDescubierta) {
            return false;
        }
    }
    return true;
}

```

esJuegoFinalizado(): Este método verifica si el jugador ha adivinado todas las letras.

actualizarLetrasDescubiertas() Este método actualiza el arreglo **descubiertas** marcando como **true** las posiciones donde el jugador ha acertado una letra.

existeLetraEnPalabra() Verifica si la letra ingresada por el jugador se encuentra en la palabra secreta. Retorna **true** si la encuentra y **false** si no.

II. PRUEBAS

¿Con qué valores comprobaste que tu práctica estuvo correcta? ¿Qué resultado esperas obtener para cada valor de entrada? ¿Qué valor o comportamiento obtuviste para cada valor de entrada?

Observando el arreglo en donde están las palabras secretas, pude contar la cantidad de espacios y compararlas a la cantidad de letras que tienen estas mismas palabras, entonces ingresando caracter por caracter, pude comprobar que estuvo correcta.

Para cada valor de entrada esperaba que me devuelva o una etapa nueva del ahorcado o, en caso acierte, las letras acertadas con los otros espacios en blanco, si es que los hubiese.

<terminated> Ahorcado [Java Application] C:\Pro

```
+---+
|
|
|
=====
- - - - -
Ingresa una letra:
o
Verificando...
- - - - o -

Ingresa una letra:
p
Verificando...
p _ _ _ o _

Ingresa una letra:
y
Verificando...
p y _ _ o _

Ingresa una letra:
t
Verificando...
p y t _ o _

Ingresa una letra:
h
Verificando...
p y t h o _

Ingresa una letra:
n
Verificando...
p y t h o n

Has adivinado la palabra.
```

Obtuve lo esperado.

	<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: 5</p>

COMMIT

Inicializar Git

1. Abre la terminal

Navega al directorio donde tienes tu proyecto:

C:\Users\santi\eclipse-workspace\FP2\src\labFP2

2. Inicializa Git

```

MINGW64/c:/Users/santi/eclipse-workspace/FP2/src/labFP2
$ cd eclipse-workspace

santi@5lofiu MINGW64 ~/eclipse-workspace
$ cd FP2

santi@5lofiu MINGW64 ~/eclipse-workspace/FP2
$ cd src

santi@5lofiu MINGW64 ~/eclipse-workspace/FP2/src
$ cd labFP2

santi@5lofiu MINGW64 ~/eclipse-workspace/FP2/src/labFP2
$ git init
Initialized empty Git repository in C:/Users/santi/eclipse-workspace/FP2/src/labFP2/.git/

```

Añadir un Commit

```

santi@5lofiu MINGW64 ~/eclipse-workspace/FP2/src/labFP2 (master)
$ git add ahorcado.java

santi@5lofiu MINGW64 ~/eclipse-workspace/FP2/src/labFP2 (master)
$ git add Ahorcado.java



santi@5lofiu MINGW64 ~/eclipse-workspace/FP2/src/labFP2 (master)
$ git commit -m "JUEGO DEL AHORCADO FP2"
[master (root-commit) 977f6ac] JUEGO DEL AHORCADO FP2
1 file changed, 147 insertions(+)
create mode 100644 Ahorcado.java

```

Vincular tu Proyecto a un Repositorio de GitHub

1. Debemos obtener la URL del repositorio .

2. Vincula tu proyecto local con el repositorio remoto en GitHub:

	<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: 6</p>

```
santi@5lofiu MINGW64 ~/eclipse-workspace/FP2/src/labFP2 (master)
$ git remote add origin https://github.com/santiagopalma12/PALMA_APAZA_SANTIAGO_ENRIQUE_LABORATORIO_02.git
git
error: remote origin already exists.

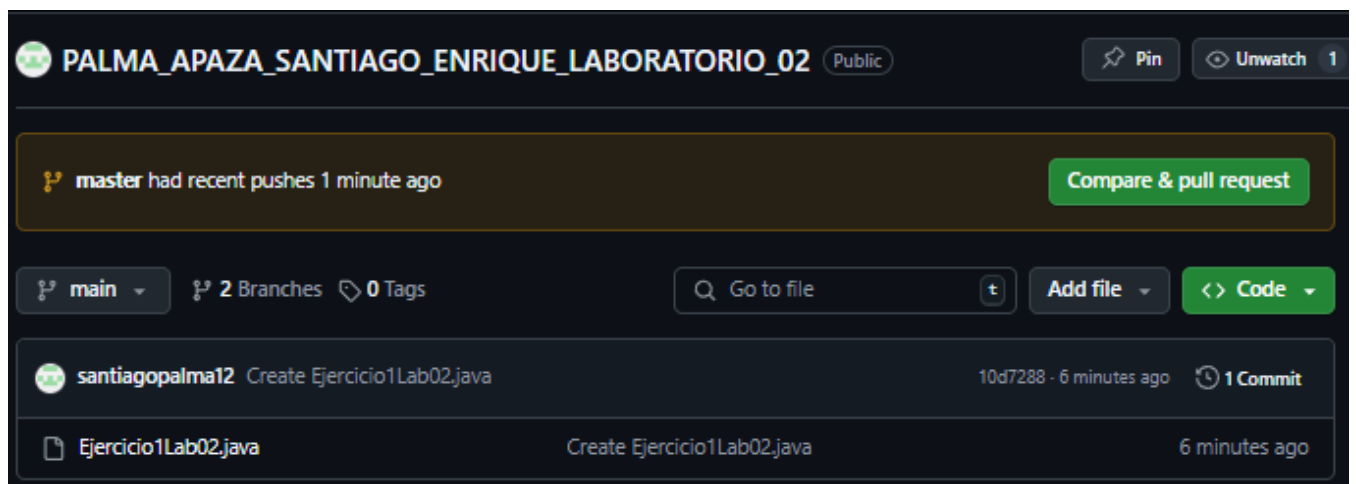
santi@5lofiu MINGW64 ~/eclipse-workspace/FP2/src/labFP2 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.23 KiB | 1.23 MiB/s, done.
Total 3 (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/santiagopalma12/PALMA_APAZA_SANTIAGO_ENRIQUE_LABORATORIO_02/pull/new/master
remote:
To https://github.com/santiagopalma12/PALMA_APAZA_SANTIAGO_ENRIQUE_LABORATORIO_02.git
 * [new branch]      master -> master

santi@5lofiu MINGW64 ~/eclipse-workspace/FP2/src/labFP2 (master)
$
```

Subir los Cambios a GitHub

Envía los cambios al repositorio remoto:

Verificar en GitHub



The screenshot shows the GitHub interface for a public repository named "PALMA_APAZA_SANTIAGO_ENRIQUE_LABORATORIO_02". At the top, there are buttons for "Pin" and "Unwatch". Below this, a notification bar indicates that the "master" branch had recent pushes 1 minute ago, with a "Compare & pull request" button. The repository has 2 branches and 0 tags. A search bar and "Go to file" button are present. Below the repository information, a commit by "santiagopalma12" is shown, titled "Create Ejercicio1Lab02.java", with a commit hash of "10d7288" and a timestamp of "6 minutes ago". The commit message "Create Ejercicio1Lab02.java" is visible, along with a "1 Commit" indicator.

iii. RUBRICA

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	4	
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	1	
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	0	
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		17	

	<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

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

Conclusiones del Juego del Ahorcado:

- El juego del ahorcado es una excelente práctica para aplicar conceptos de manipulación de cadenas y estructuras de control en programación.
- Permite mejorar la lógica de validación y manejo de errores, ya que se deben controlar las entradas del usuario y actualizar el estado del juego en función de sus aciertos o fallos.
- Es un ejemplo claro de cómo dividir un problema complejo en tareas más pequeñas y manejables, como la selección de palabras, la actualización de las pistas y el control de las etapas del ahorcado.

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.

Metodología Aplicada:

- El problema se dividió en varios métodos pequeños, cada uno con una responsabilidad clara, como ingresar una letra, actualizar el tablero, y verificar el estado del juego.
- Se utilizó un bucle para repetir las jugadas hasta que se completaran todas las letras o se agotaran los intentos.
- Se implementaron controles para garantizar que el usuario ingrese letras válidas.

REFERENCIAS Y BIBLIOGRAFÍA

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

Aedo López, Marco Wilfredo. Fundamentos de programación I: Java Básico. Universidad Nacional de San Agustín, 1ª ed., 31 de julio de 2019, ISBN 978-612-4337-55-0, 116 páginas, impreso, tapa blanda o rústica.