
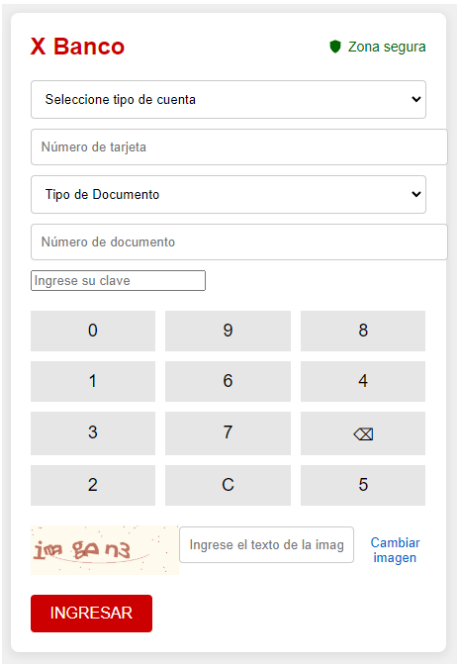


	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 1

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	PROGRAMACIÓN WEB 2 – Grupo B				
TÍTULO DE LA PRÁCTICA:					
NÚMERO DE PRÁCTICA:	<i>03</i>	AÑO LECTIVO:	<i>2024 – B</i>	NRO. SEMESTRE:	<i>03</i>
FECHA DE PRESENTACIÓN	<i>11/10/2024</i>	HORA DE PRESENTACIÓN	<i>23:59</i>		
INTEGRANTE (s) <i>NIKOLE VALERY SALAS IDME</i>				NOTA (0-20)	
DOCENTE(s): • <i>Mg. Ing. Lino Pinto Oppe</i>					

RESULTADOS Y PRUEBAS	
I. URL AL DIRECTORIO ESPECÍFICO:	<p><i>https://github.com/LINOPINTO2023/PWII-2024B/tree/Grupo-Miau/LABORATORIO%20III%20SALAS_IDME_NIKOLE</i></p>
II. EJERCICIOS RESUELTOS:	<p>Ejercicio 01: Cree un teclado random para banca por internet.</p> <div data-bbox="592 1417 1050 2078">  </div>

	<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: 2</p>

1. PORCIONES DE CÓDIGO FUENTE IMPORTANTES:

1. Validación y actualización del CAPTCHA: Importancia: Esta sección se encarga de generar, mostrar y validar el CAPTCHA, lo cual es fundamental para garantizar la seguridad del formulario. Asegura que solo los usuarios que ingresen el texto correcto puedan continuar, evitando posibles bots o accesos no autorizados. (editado)

```

19 // Función para obtener una imagen CAPTCHA aleatoria
20 function getRandomCaptcha() {
21   const randomIndex = Math.floor(Math.random() * captchaImages.length);
22   const selectedCaptcha = captchaImages[randomIndex];
23   currentCaptchaText = selectedCaptcha.text;
24   return captchaFolder + selectedCaptcha.src;
25 }
26
27 function updateCaptcha() {
28   captchaImageElement.src = getRandomCaptcha();
29 }
30
31 refreshCaptchaButton.addEventListener('click', updateCaptcha);
32
33 document.addEventListener('DOMContentLoaded', () => {
34   updateCaptcha();
35 });
36
37 function validateCaptcha() {
38   const userCaptchaInput = captchaInputElement.value.trim();
39
40   if (userCaptchaInput === currentCaptchaText) {
41     alert('Bienvenido Usuario');
42     location.reload();
43   } else {
44     alert('CAPTCHA incorrecto, por favor intente de nuevo');
45     updateCaptcha();
46   }
47 }

```

2. Teclado virtual: Importancia: El teclado virtual permite a los usuarios ingresar su PIN de manera segura sin usar un teclado físico, reduciendo el riesgo de keyloggers o ataques similares. Además, garantiza que el PIN tenga una longitud máxima de 6 dígitos y proporciona botones para borrar o reiniciar el campo.

```

61 function renderKeyboard() {
62   keyboard.innerHTML = "";
63   shuffleArray(keys);
64   keys.forEach(key => {
65     const button = document.createElement('button');
66     button.textContent = key;
67     button.className = 'key';
68     button.onclick = () => handleKeyPress(key);
69     keyboard.appendChild(button);
70   });
71 }
72
73 function handleKeyPress(key) {
74   if (key === 'C') {
75     pinInput.value = "";
76   } else if (key === '⌫') {
77     pinInput.value = pinInput.value.slice(0, -1);
78   } else if (pinInput.value.length < 6) {
79     pinInput.value += key;
80   }
81 }

```

2. COMMITS :

imagenes	Verified	875b88d		
arreglar errores	Verified	4389c6d		
pulsaciones de teclas	Verified	6363f45		
Funciones para mezclar y renderizar teclado	Verified	3bf7749		
validar captcha y envío	Verified	5480c38		
actualizar captcha y clic	Verified	82aa763		
obtener imagen captcha aleatoria	Verified	c331a67		
definir teclas y captcha	Verified	07a3c3b		
obtener elementos del DOM	Verified	7b6e5da		
agregar hoja de estilos	Verified	ed72736		
captcha y botón de ingreso	Verified	59f1496		
campos de entrada	Verified	691a881		
Agregar cabecera y selección de cuenta	Verified	11ac769		
estructura basica	Verified	2335483		

Ejercicio 02: Cree una calculadora básica como la de los sistemas operativos, que pueda utilizar la función eval() y que guarde todos las operaciones en una pila. Mostrar la pila al pie de la página web.

0

mod

(

)

⌫

x²

√

π

÷

7

8

9

×

4

5

6

-

1

2

3

+



0

.

C

=

Historial de Operaciones

	<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: 4</p>

1. PORCIONES DE CÓDIGO FUENTE IMPORTANTES:

1. calculateResult() Importancia: Esta función maneja el cálculo final de la expresión ingresada por el usuario, reemplazando operadores especiales como " π " y " $\sqrt{\quad}$ " con sus equivalentes en JavaScript. Además, gestiona posibles errores durante el cálculo y guarda el resultado en el historial. Es esencial para la funcionalidad de la calculadora, ya que ejecuta los cálculos y presenta los resultados. (editado)

```

72 function calculateResult() {
73   if (currentExpression === "") return;
74   try {
75     let expressionToEvaluate = currentExpression
76       .replace(/pi/g, 'Math.PI')
77       .replace(/\^/g, 'Math.sqrt()')
78       .replace(/\^2/g, '**2')
79       .replace(/%/g, '/100');
80     .replace(/mod/g, '%');
81
82     const result = eval(expressionToEvaluate);
83
84     const roundedResult = Number(result.toFixed(10));
85
86     const operation = `${displayExpression} = ${roundedResult}`;
87     operationStack.push(operation);
88     updateHistory();
89     currentExpression = roundedResult.toString();
90     displayExpression = roundedResult.toString();
91   } catch (error) {
92     currentExpression = 'Error';
93     displayExpression = 'Error';
94   }
95 }

```

2. Manejo de eventos en los botones: Importancia: Esta función maneja las acciones especiales de la calculadora, como operadores matemáticos, borrar, limpiar y calcular el resultado. Es esencial para procesar las funciones avanzadas de la calculadora, como raíces cuadradas, potencias, porcentajes y operadores básicos, permitiendo que el usuario realice más que simples operaciones numéricas.

```

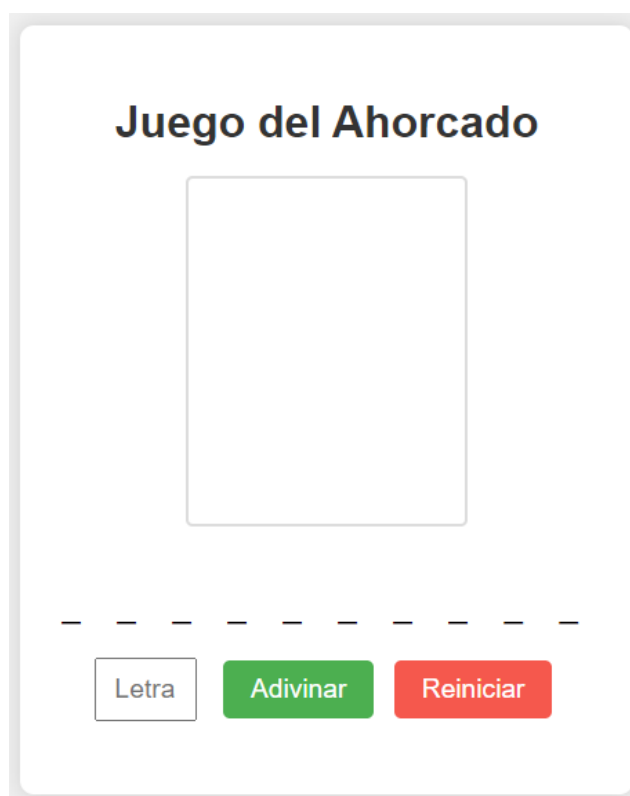
26 // Manejar acciones (operadores y funciones especiales)
27 function handleAction(action) {
28   switch (action) {
29     case '+':
30     case '-':
31     case '*':
32     case '/':
33     appendToExpression(action);
34     break;
35     case 'mod':
36     appendToExpression(' mod ');
37     break;
38     case '^':
39     calculateResult();
40     break;
41     case 'clear':
42     clearCalculator();
43     break;
44     case 'delete':
45     deleteLastChar();
46     break;
47     case 'pi':
48     appendToExpression('pi');
49     break;
50     case '^':
51     appendToExpression('^');
52     break;
53     case '^2':
54     appendToExpression('^2');
55     break;
56     case 'percent':
57     handlePercent();
58     break;
59     case '(':
60     case ')':
61     appendToExpression(action);
62     break;
63   }
64 }



```

2. COMMITS:

Implementar función para agregar a la expresión	Verified	963cb18			
Implementar función para manejar acciones	Verified	b4f29f3			
definir variables y evento de clic	Verified	2c41142			
obtener elementos del DOM	Verified	01a2f9a			
arreglar errores y terminar	Verified	88639ed			
añadir el script	Verified	d5da5be			
agregar seccion de historial	Verified	a6f1b6a			
agregar botones a la calculadora	Verified	a195c48			
estructura de la calculadora	Verified	ff178fe			
enlaces a estilos	Verified	18bb5db			
estructura básica del html	Verified	d36b148			

Ejercicio 03: Cree una versión de el juego 'el ahorcado' que grafique con canvas paso a paso desde el evento onclick() de un botón.



	<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>

1. PORCIONES DE CÓDIGO FUENTE IMPORTANTES:

1. Importancia: Esta función inicializa el juego seleccionando una palabra aleatoria, creando un arreglo de guiones bajos para mostrar el progreso del jugador y reiniciando el lienzo y los intentos. Es fundamental para establecer las condiciones iniciales en cada partida.

```

15 // Inicializa el juego
16 function iniciarJuego() {
17     palabraSeleccionada = palabras[Math.floor(Math.random() * palabras.length)];
18     palabraAdivinada = Array(palabraSeleccionada.length).fill('_');
19     intentosIncorrectos = 0;
20     actualizarPalabra();
21     limpiarLienzo();
22     mensaje.textContent = "";
23     btnAdivinar.disabled = false;
24 }



```

2. Importancia: Dibuja el ahorcado de manera progresiva a medida que el jugador comete errores. Visualmente, da retroalimentación sobre el avance del juego y actúa como una representación gráfica del límite de errores permitidos.

```

36 // Dibuja el ahorcado en el lienzo
37 function dibujarAhorcado() {
38     contexto.strokeStyle = "#333";
39     contexto.lineWidth = 2;
40     switch (intentosIncorrectos) {
41         case 1:
42             // Base
43             contexto.beginPath();
44             contexto.moveTo(20, 230);
45             contexto.lineTo(180, 230);
46             contexto.stroke();
47             break;
48         case 2:
49             // Poste vertical
50             contexto.beginPath();
51             contexto.moveTo(40, 230);
52             contexto.lineTo(40, 20);
53             contexto.stroke();
54             break;
55         case 3:
56             // Poste horizontal
57             contexto.beginPath();
58             contexto.moveTo(40, 20);
59             contexto.lineTo(100, 20);
60             contexto.stroke();
61             break;
62         case 4:
63             // Cuerda
64             contexto.beginPath();
65             contexto.moveTo(100, 20);
66             contexto.lineTo(100, 50);
67             contexto.stroke();
68             break;
69         case 5:
70             // Cabeza
71             contexto.beginPath();
72             contexto.arc(100, 70, 20, 0, Math.PI * 2);
73             contexto.stroke();
74             break;
75         case 6:
76             // Cuerpo, brazos y piernas
77             contexto.beginPath();
78             contexto.moveTo(100, 90);
79             contexto.lineTo(100, 150);
80             contexto.moveTo(100, 110);
81             contexto.lineTo(60, 100); // Brazo izquierdo
82             contexto.moveTo(100, 110);
83             contexto.lineTo(140, 100); // Brazo derecho
84             contexto.moveTo(100, 150);
85             contexto.lineTo(70, 190); // Pierna izquierda
86             contexto.moveTo(100, 150);
87             contexto.lineTo(130, 190); // Pierna derecha
88             contexto.stroke();
89             break;
90     }
91 }

```

	<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>

3. Importancia: Esta es la función que maneja la lógica central del juego. Valida la letra ingresada, verifica si está en la palabra, actualiza el estado del juego (ya sea mostrando letras correctas o incrementando los errores) y controla la finalización del juego cuando el jugador gana o pierde. La interactividad del jugador con el juego depende de esta función.

```

93 // Maneja el intento de adivinar una letra
94 function hacerAdivinanza() {
95     const letra = inputLetra.value.toLowerCase();
96     if (letra.length !== 1 || ![a-zñ]/.test(letra)) {
97         mensaje.textContent = 'Por favor, ingresa una letra válida.';
98         return;
99     }
100
101     if (palabraSeleccionada.includes(letra)) {
102         for (let i = 0; i < palabraSeleccionada.length; i++) {
103             if (palabraSeleccionada[i] === letra) {
104                 palabraAdivinada[i] = letra;
105             }
106         }
107         actualizarPalabra();
108         if (!palabraAdivinada.includes('_')) {
109             mensaje.textContent = '¡Felicidades! Has ganado.';
110             btnAdivinar.disabled = true;
111         }
112     } else {
113         intentosIncorrectos++;
114         dibujarAhorcado();
115         if (intentosIncorrectos === maxIntentos) {
116             mensaje.textContent = 'Juego Terminado. La palabra era: ' + palabraSeleccionada;
117             btnAdivinar.disabled = true;
118         }
119     }
120
121     inputLetra.value = '';
122     inputLetra.focus();
123 }

```

2. COMMITS:

arreglar errores	Verified	56328d5		
Manejar eventos de reinicio y adivinanza	Verified	c231a79		
Manejar intento de adivinar letra	Verified	278a311		
dibujar ahorcado	Verified	4682c6d		
actualizar palabra y limpiar lienzo	Verified	7372f63		
inicializar juego	Verified	8a219e8		
Definición de variables y elementos del DOM	Verified	0dc4d8b		
hoja de estilos	Verified	463275c		
arreglar errores	Verified	4d85fca		
agregar script	Verified	2769bda		
agregar botones	Verified	54d8b7c		
Agregar canvas y elementos de entrada	Verified	4a21861		
contenedor de juego y referencia	Verified	e10289e		
estructura basica	Verified	8e78288		

III. EJECUCIONES

127.0.0.1:3000 says
Bienvenido Usuario

OK

X Banco 🛡️ Zona segura

Debito

1029390123812903

DNI

72932839

1	0	7
2	6	✖
4	9	C
8	3	5

Prueba 1 [Cambiar imagen](#)

INGRESAR

1.2857142857

mod	()	✖
x²	√	π	÷
7	8	9	×
4	5	6	-
1	2	3	+
0	.	C	
=			

Historial de Operaciones

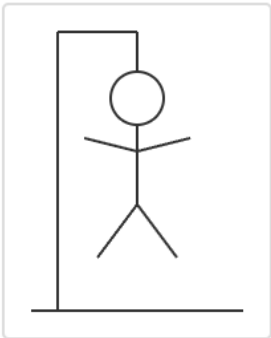
9/7 = 1.2857142857

100^2/√(55) = 1348.3997249265

5 mod 10 = 5

5*(9+2) = 55

Juego del Ahorcado



— — — — — p —

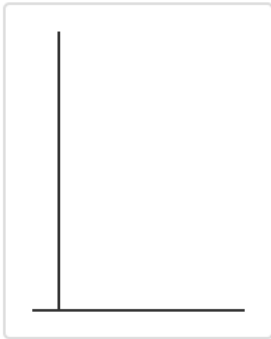
Letra

Adivinar

Reiniciar

Juego Terminado. La palabra era: javascript

Juego del Ahorcado



t e c n o l o g i a

Letra

Adivinar

Reiniciar

¡Felicidades! Has ganado.

CUESTIONARIO

1. Explique una herramienta para ofuzcar código JavaScript.

La ofuscación de código JavaScript es el proceso de hacer que el código original sea más difícil de entender o modificar, sin cambiar su funcionamiento. Esto es útil en producción, ya que ayuda a proteger el código de que otros lo lean o lo alteren fácilmente. Además, muchas herramientas de ofuscación también minifican el código, lo que significa que eliminan espacios, comentarios y otros elementos innecesarios para reducir su tamaño.

Una herramienta popular para ofuscar y minificar código JavaScript es UglifyJS. Esta herramienta no solo reduce el tamaño del archivo, sino que también cambia los nombres de las variables y elimina código innecesario para optimizarlo.

Muestre un ejemplo de su uso en uno de los ejercicios de la tarea.

- script_ejercicio_01.js (development).
- script_ejercicio_01.min.js (production).



	<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: 10</p>

Tabla 2: Niveles de desempeño

Puntos	Nivel			
	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

Tabla 3: Rúbrica para contenido del Informe y evidencias

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error o omisión)	4		1	
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4		2	
3. Ejecución	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	4		3	
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2		1	
7. Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2		2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados a partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4		3	
Total		20		12	