

Informe de Laboratorio 03

Tema: JavaScript

Nota

Estudiante	Escuela	Asignatura
Fernandez Huarca, Rodrigo Alexander rfernandezh@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 20230465

Laboratorio	Tema	Duración
03	JavaScript	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 16 Mayo 2024	Al 19 Mayo 2024

1. Tarea

- Programar en JavaScript sobre una página web html básica.
- Ejercicio 01: Cree un teclado random para banca por internet.
- 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.
- Ejercicio 03: Cree una versión de el juego 'el ahorcado' que grafique con canvas paso a paso desde el evento onclick() de un botón.

2. Acceso al Repositorio en GitHub

- URL para el laboratorio 03 para clonar o recuperar.
 - <https://github.com/RdrigoFH/FernandezHuarcaRodrigoAlexander.git>
- URL para el laboratorio 03 en el Repositorio GitHub.
 - <https://github.com/RdrigoFH/FernandezHuarcaRodrigoAlexander>

3. Ejercicios propuestos

3.1. Ejercicio 1

- Para esta sección se creó una estructura *html* básica, sobre la que se añadieron

Listing 1: index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Banca por Internet</title>
7    <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10   <div class="container">
11     
12     <h2>Banca por internet</h2>
13
14     <div id="element">
15       <h3>Numero de Tarjeta</h3>
16       <input type="number" id="tarjeta">
17     </div>
18
19     <div id="element">
20       <h3>Documento de Identidad</h3>
21       <input type="number" id="dni">
22     </div>
23
24     <h3>Ingresa su contraseña</h3>
25     <input type="password" id="display" class="display">
26     <div id="keypad" class="keypad"></div>
27
28     <button id="login">Ingresar</button>
29   </div>
30
31   <script src="script.js"></script>
32 </body>
33 </html>
```

- Estilos aplicados a los componentes del *html*

Listing 2: styles.css

```
1 .container{
2     display: flex;
3     flex-direction: column;
4     align-items: center;
5 }
6
7 .display {
8     width: 225px;
9     margin: 20px auto;
10    padding: 10px;
11    font-size: 24px;
12    text-align: right;
13    border: 1px solid #ccc;
14    border-radius: 5px;
15    display: block;
16    background-color: #f9f9f9;
17 }
18
19 .keypad {
20     display: grid;
21     grid-template-columns: repeat(3, 1fr);
22     gap: 10px;
23     max-width: 200px;
24     margin: auto;
25     padding: 20px;
26     border: 1px solid #ccc;
27     border-radius: 10px;
28     background-color: #f9f9f9;
29 }
30
31 .key {
32     display: flex;
33     justify-content: center;
34     align-items: center;
```

```
35 width: 60px;
36 height: 60px;
37 font-size: 24px;
38 font-weight: bold;
39 background-color: #fff;
40 border: 1px solid #ccc;
41 border-radius: 5px;
42 cursor: pointer;
43 user-select: none;
44 }
45
46 .key:active {
47   background-color: #e0e0e0;
48 }
49 #img{
50   margin: -10px;
51 }
52 #element{
53   display: flex;
54   align-items: center;
55 }
56 input{
57   height: 30px;
58   width: 300px;
59   margin: 15px;
60 }
61 h3 {
62   margin: 5px;
63 }
64 #login{
65   border-radius: 15px;
66   border: none;
67   background-color: black;
68   color: aliceblue;
69   padding: 15px;
70   margin-top: 10px;
71 }
```

- JavaScript usado para mostrar un teclado numérico desordenado generado por la función `shuffle()` y luego mostrado en pantalla mediante un `foreach` que fue creando los elementos.

Listing 3: script.js

```
1 document.addEventListener('DOMContentLoaded', () => {
2   const keypad = document.getElementById('keypad');
3   const display = document.getElementById('display');
4
5   function shuffle(array) {
6     for (let i = array.length - 1; i > 0; i--) {
7       const j = Math.floor(Math.random() * (i + 1));
8       [array[i], array[j]] = [array[j], array[i]];
9     }
10    return array;
11  }
12
13  const numbers = shuffle(Array.from({ length: 10 }, (_, i) => i));
14
15  numbers.forEach(number => {
16    const key = document.createElement('div');
17    key.classList.add('key');
18    key.textContent = number;
19    key.addEventListener('click', () => {
20      display.value += number;
21    });
22    keypad.appendChild(key);
23  });
24 });
```

Figura 1: Resultados/Pagina ejercicio 01




Banca por internet

Numero de Tarjeta

Documento de Identidad

Ingrese su contraseña

1	8	2
4	9	6
3	7	5
0		

Ingresar

3.2. Ejercicio 2

- En este documento *html* Se empezaron creando elementos `<div>` para cada fila de nuestra calculadora, se agregaron las funciones directamente a cada botón, en este caso la función usada fue `show(value)` que mostraba el valor en el text field display.

Listing 4: index.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Calculator</title>
7   <link rel="stylesheet" href="styles.css">
8
9 </head>
10 <body>
11   <div class="container">
12     <div class="calculator">
13       <form>
14         <div class="display">
15           <input type="text" name="" display="" id="display">
16         </div>
17       <div>

```

```
18         <input type="button" value="AC" class="sign"
    onclick="display.value = ' '>
19         <input type="button" value="&lt;" class="sign"
    onclick="display.value = display.value.toString().slice(0,-1)">
20         <input type="button" value="mod" class="sign"
    onclick="display.value += '%'">
21         <input type="button" value="/" class="sign"
    onclick="display.value += '/'">
22     <div>
23         <input type="button" value="7" onclick="show('7')">
24         <input type="button" value="8" onclick="show('8')">
25         <input type="button" value="9" onclick="show('9')">
26         <input type="button" value="X" class="sign"
    onclick="display.value += '*'">
27     </div>
28     <div>
29         <input type="button" value="4" onclick="show('4')">
30         <input type="button" value="5" onclick="show('5')">
31         <input type="button" value="6" onclick="show('6')">
32         <input type="button" value="+" class="sign"
    onclick="display.value += '+'">
33     </div>
34     <div>
35         <input type="button" value="1" onclick="show('1')">
36         <input type="button" value="2" onclick="show('2')">
37         <input type="button" value="3" onclick="show('3')">
38         <input type="button" value="-" class="sign"
    onclick="display.value += '-'">
39     </div>
40     <div>
41         <input type="button" value="0" onclick="show('0')">
42         <input type="button" value="," onclick="display.value
    += ','">
43         <input type="button" value="=" class="eq"
    onclick="calculate()">
44     </div>
45 </form>
46 </div>
47 </div>
48 <div class="history">
```

```
49     <h3 id="history" >"History" <br></h3>
50   </div>
51   <script src="index.js"></script>
52 </body>
53 </html>
```

- Plantilla CSS usada para darle estilos a la calculadora, se opto por un estilo moderno y se coloco la tabla de historial al lado derecho.

Listing 5: styles.css

```
1  *{
2    margin: 0;
3    padding: 0;
4    font-family: Arial, sans-serif;
5    box-sizing: border-box;
6  }
7  .container{
8    width: 100%;
9    height: 100vh;
10   background: #555656;
11   display: flex;
12   align-items: center;
13   justify-content: center;
14 }
15 .calculator{
16   background: #3d4c56;
17   padding: 20px;
18   border-radius: 10px;
19 }
20 .history{
21   background: #ffffff;
22   padding: 10px;
23   border-radius: 10px;
24   margin-left: 50px;
25   height: 400px;
26   width: 250px;
27 }
28 .calculator form input{
29   border: 0;
30   outline: 0;
31   width: 50px;
```



```
32 height: 50px;
33 border-radius: 5px;
34 margin: 5px;
35 cursor: pointer;
36 }
37 form .display{
38 display: flex;
39 justify-content: flex-end;
40 margin: 20px 0;
41 }
42 form .display input{
43 text-align: right;
44 flex: 1;
45 font-size: 35px;
46 }
47 form input.eq {
48 width: 110px;
49 border-radius: 50px;
50 }
51 form input.sign {
52 border-radius: 50px;
53 background-color: rgb(219, 170, 22);
54 }
55 input:hover{
56 background-color: #697e99a7;
57 }
```

- Javascript que selecciona los elemento display e histor, primero mediante la dfuncion show, semostraran los elementos en la pantalla de la calculadora(display) y luego mediante caluculate() que se activara cuando se de chlick sobre el botón de igual, se calcularan los valores y se almacenaran en hitory y se mostraran en display.

Listing 5: script.js

```
const display = document.getElementById("display");
const history = document.getElementById("history")
function show(input){
    display.value += input;
}
function calculate(){
    history.innerHTML+= display.value + "=" + eval(display.value) + '<br>';
    display.value = eval(display.value);
}
```

■ Resultados Finales

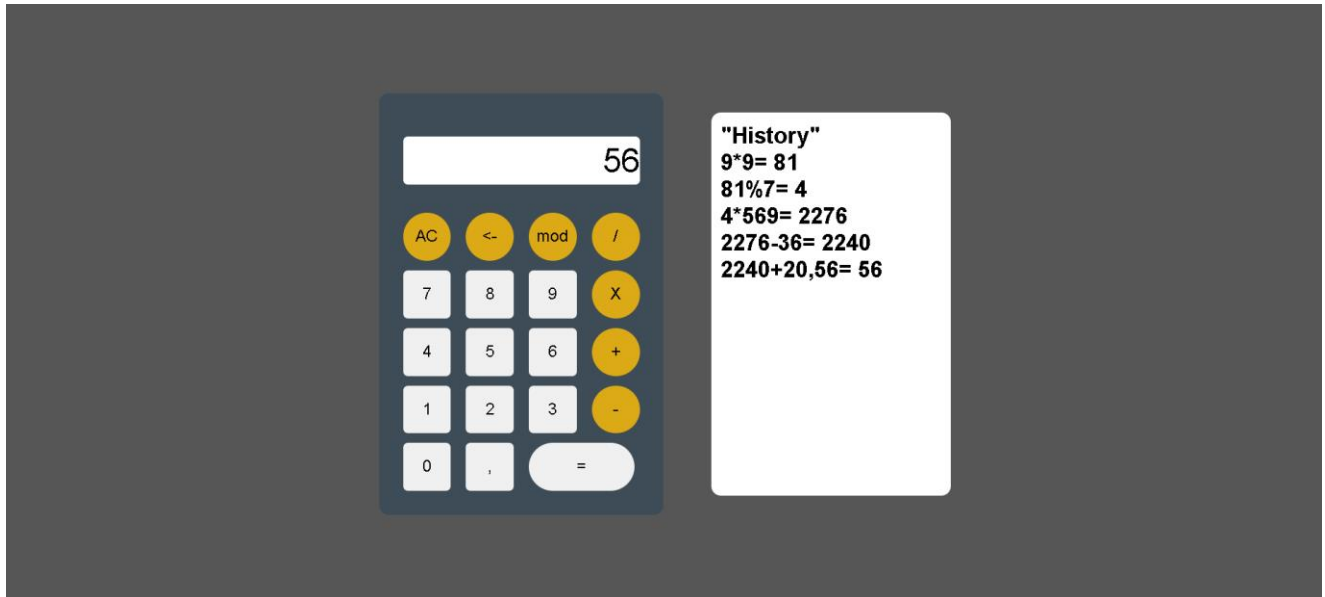


Figura 2: Resultados/Pagina Ejercicio02

3.3. Ejercicio 3

- Estructura del *html* que contendrá el título del juego y los intentos restantes, el teclado se agregará en el *script.js* al igual que la app del teclado virtual del Ejercicio01, lo mismo sucede con el stick man del hombre ahorcado.

Listing 7: index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Hangman Game</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10  <div id="game-container">
11    <h1>Hangman Game</h1>
12    
13    <p id="word-display"></p>
14    <p id="message"></p>
15    <div id="letters-container"></div>
16    <p id="remaining-attempts">Remaining Attempts: <span>
```

```
17 id="attempts">6</span></p>
18     <button id="restart-btn" style="display: none;">Restart</button>
19 </div>
20
21 <script src="script.js"></script>
22 </body>
    </html>
```

- JavaScript que primero crea el teclado alfabético donde cada letra tendrá el listener `handleGuess(letter)` y adjunta la imagen 0, esto mediante la función `initializeGame()`, seguidamente se maneja la lógica del juego mediante la función `handleGuess`.

Listing 8: script.js

```
1 document.addEventListener('DOMContentLoaded', () => {
2     const words = ['ejemplo', 'programacion web', 'programming', 'developer',
3 'computer'];
4     let selectedWord = '';
5     let displayedWord = '';
6     let attempts = 6;
7     let guessedLetters = [];
8
9     const wordDisplay = document.getElementById('word-display');
10    const messageDisplay = document.getElementById('message');
11    const lettersContainer = document.getElementById('letters-container');
12    const attemptsDisplay = document.getElementById('attempts');
13    const restartBtn = document.getElementById('restart-btn');
14    const hangmanImage = document.getElementById('hangman-image');
15
16    function initializeGame() {
17        selectedWord = words[Math.floor(Math.random() * words.length)];
18        displayedWord = '_'.repeat(selectedWord.length);
19        attempts = 6;
20        guessedLetters = [];
21
22        wordDisplay.textContent = displayedWord.split('').join(' ');
23        messageDisplay.textContent = '';
24        attemptsDisplay.textContent = attempts;
25        lettersContainer.innerHTML = '';
```

```
26     hangmanImage.src = 'images/hangman-0.svg';
27
28     for (let i = 0; i < 26; i++) {
29         const letter = String.fromCharCode(65 + i);
30         const letterButton = document.createElement('span');
31         letterButton.textContent = letter;
32         letterButton.classList.add('letter');
33         letterButton.addEventListener('click', () => handleGuess(letter));
34         lettersContainer.appendChild(letterButton);
35     }
36
37     restartBtn.style.display = 'none';
38 }
39
40 function handleGuess(letter) {
41     if (guessedLetters.includes(letter)) return;
42
43     guessedLetters.push(letter);
44     const letterButtons = document.querySelectorAll('.letter');
45     letterButtons.forEach(btn => {
46         if (btn.textContent === letter) btn.classList.add('disabled');
47     });
48
49     if (selectedWord.toUpperCase().includes(letter)) {
50         let newDisplayedWord = '';
51         for (let i = 0; i < selectedWord.length; i++) {
52             if (selectedWord[i].toUpperCase() === letter) {
53                 newDisplayedWord += selectedWord[i];
54             } else {
55                 newDisplayedWord += displayedWord[i];
56             }
57         }
58         displayedWord = newDisplayedWord;
59         wordDisplay.textContent = displayedWord.split('').join(' ');
60         if (!displayedWord.includes('_')) {
61             messageDisplay.textContent = 'Congratulations! You won!';
62             endGame();
63         }
64     } else {
65         attempts--;
```

```
66     attemptsDisplay.textContent = attempts;
67     hangmanImage.src = `images/hangman-${6 - attempts}.svg`;
68     if (attempts === 0) {
69         messageDisplay.textContent = `Game over! The word was
    "${selectedWord}".`;
70         endGame();
71     }
72 }
73 }
74
75 function endGame() {
76     const letterButtons = document.querySelectorAll('.letter');
77     letterButtons.forEach(btn => btn.classList.add('disabled'));
78     restartBtn.style.display = 'block';
79 }
80
81 restartBtn.addEventListener('click', initializeGame);
82
83 initializeGame();
84 });
```

- Hoja de estilos para maximizar la experiencia de los usuarios con una interfaz minimalista y fácil de entender.

Listing 9: styles.css

```
1 body {
2     font-family: Arial, sans-serif;
3     text-align: center;
4     background-color: #535252;
5     margin: 0;
6     padding: 20px;
7 }
8
9 #game-container {
10     max-width: 600px;
11     margin: 0 auto;
12     background: white;
13     padding: 20px;
14     border-radius: 10px;
15     box-shadow: 0 0 10px rgba(0,0,0,0.1);
```

```
16 }
17
18 #word-display {
19     font-size: 2em;
20     letter-spacing: 0.1em;
21 }
22
23 #letters-container {
24     margin: 20px 0;
25 }
26
27 .letter {
28     display: inline-block;
29     margin: 5px;
30     padding: 10px 15px;
31     font-size: 1.5em;
32     background: #00ffd9;
33     color: white;
34     border-radius: 5px;
35     cursor: pointer;
36 }
37
38 .letter.disabled {
39     background: #cccccc;
40     cursor: not-allowed;
41 }
42
43 #remaining-attempts {
44     margin-top: 20px;
45     font-size: 1.2em;
46 }
47
48 #restart-btn {
49     margin-top: 20px;
50     padding: 10px 20px;
51     font-size: 1em;
52     background: #000000;
53     color: white;
54     border: none;
55     border-radius: 5px;
```

```
56  cursor: pointer;
57  }
58  #hangman-image {
59    max-width: 100%;
60    height: auto;
61    margin: 20px 0;
62  }
63
```

■ Resultados finales

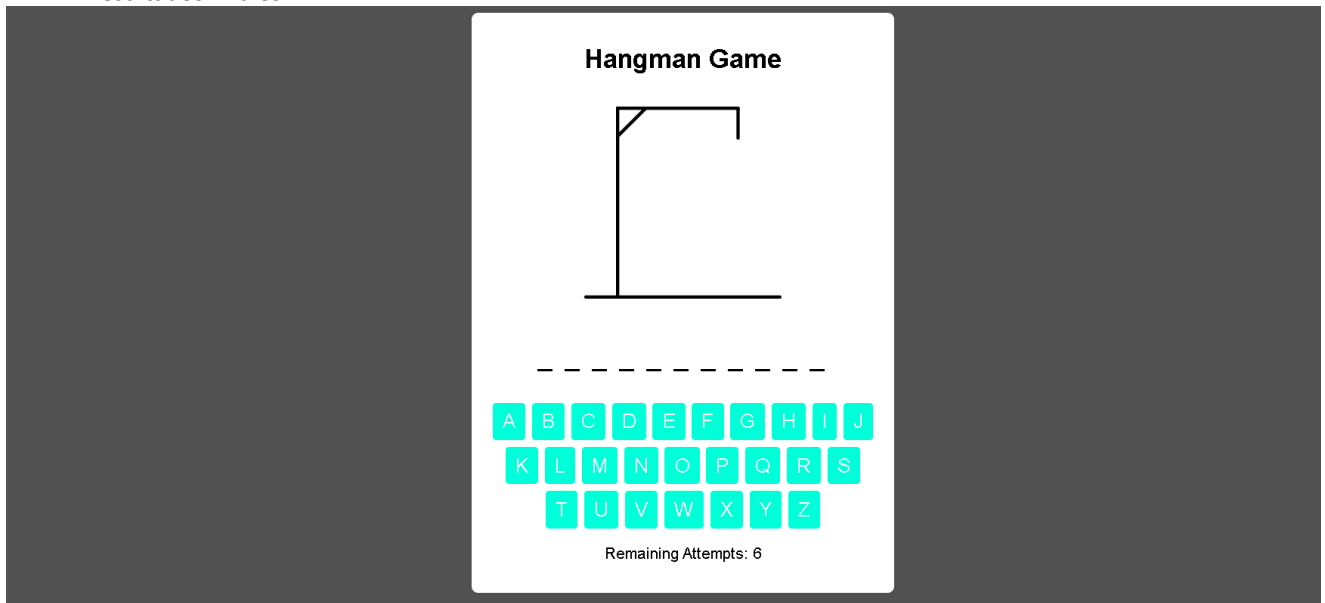


Figura 3: Resultados/Pagina Ejercicio03

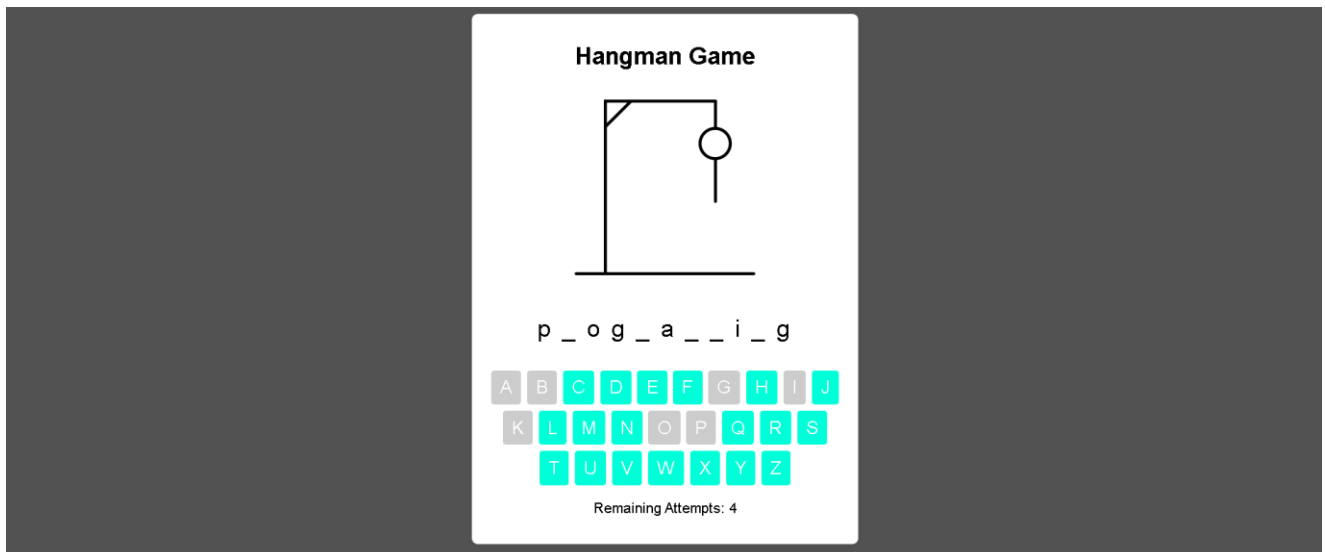


Figura 4: Resultados/Pagina Ejercicio03



Figura 5: Resultados/Pagina Ejercicio03

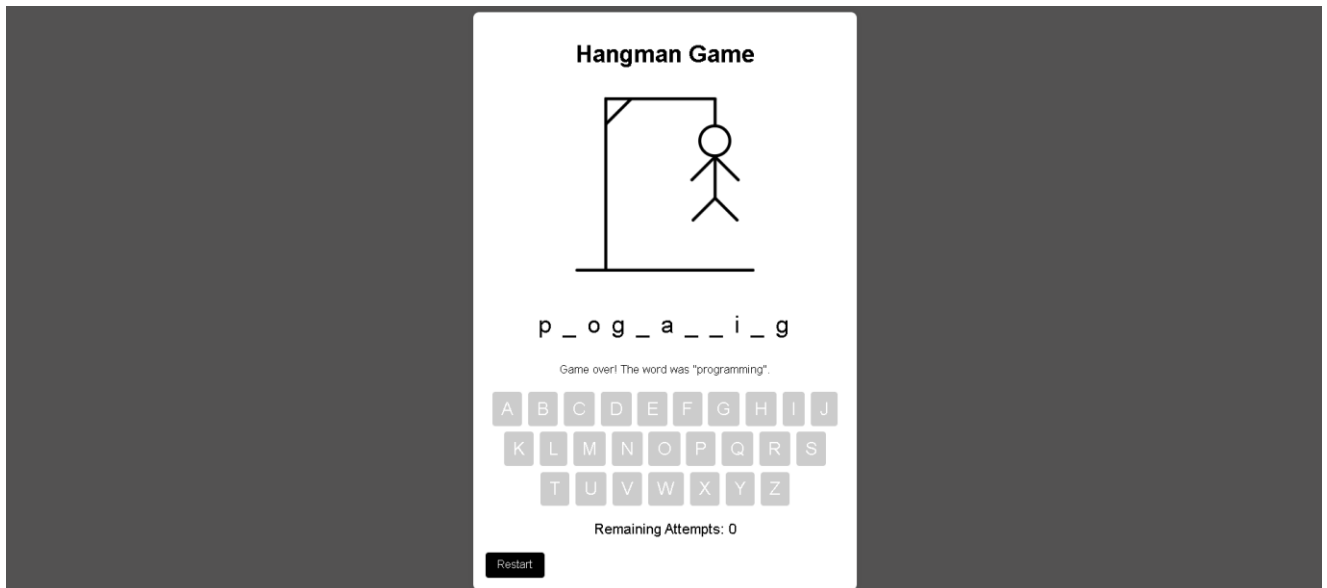


Figura 6: Resultados/Pagina Ejercicio03

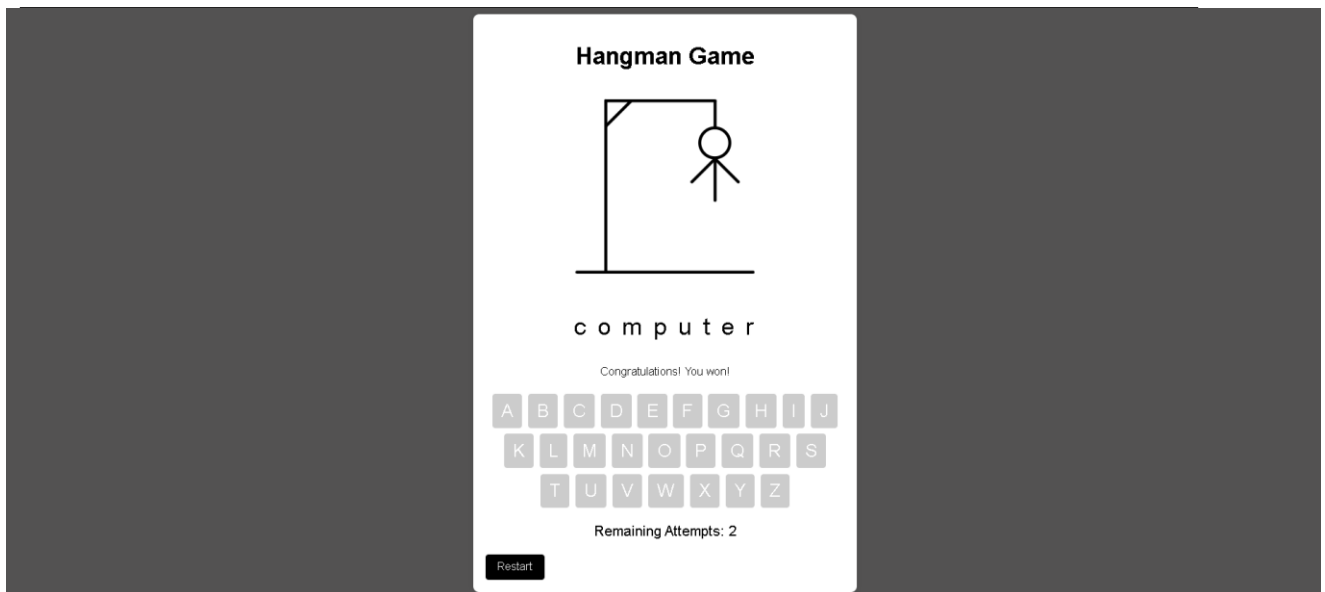


Figura 7: Resultados/Pagina Ejercicio03

4. Pregunta

4.1. Explique una herramienta para ofuzcar código JavaScript.

- <http://dean.edwards.name/packer/> Es una herramienta de ofuscación de JavaScript es un programa que toma código JavaScript legible y lo transforma en un código más difícil de entender para los humanos. Esto se logra mediante la aplicación de diversas técnicas

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

Ejercicio 01:

```
document.addEventListener('DOMContentLoaded', ()=>{const
keypad=document.getElementById('keypad');const
display=document.getElementById('display');function shuffle(array){for(let i=array.length-
1;i>0;i--){const j=Math.floor(Math.random()*(i+1));[array[i],array[j]]=
[array[j],array[i]]}return array}const numbers=shuffle(Array.from({length:10},
(_,i)=>i));numbers.forEach(number=>{const
key=document.createElement('div');key.classList.add('key');key.textContent=number;key.addEvent
Listener('click',()=>{display.value+=number});keypad.appendChild(key)}});
```

4.3. Adjunte a su repositorio ambas versiones:

- Versiones implementadas en el repositorio

5. Rubrica

5.1. Sobre el informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumplió con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	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 demostración

	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	2	
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	1.5	
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	2	
Total		20		15.5	

6. Referencias

- <https://www.w3schools.com/js/>
- <https://learnjavascript.online/>