

Informe de Laboratorio 03

Tema: JavaScript

Nota

Estudiante	Escuela	Asignatura
David Alfredo Huamani Ollachica dhuamano@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 1702122

Laboratorio	Tema	Duración
03	JavaScript	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 17 Mayo 2024	Al 18 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 todas 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. Equipos, materiales y temas utilizados

- Sistema Operativo Ubuntu GNU Linux 23 lunar 64 bits Kernell 6.2.
- VIM 9.0.
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.
- HTML, CSS y JavaScript
- DOM

3. URL de Repositorio Github

- URL para el laboratorio 03 en el Repositorio GitHub.
- https://github.com/dev1d123/pw2_lab03

```
PS C:\Users\Windows\Desktop\u\pw2_lab03> git log
commit 81de50f3068a713c9f712dae2f869ca123fb9629 (HEAD -> main, origin/main, origin/HEAD)
Author: dev1d123 <139636102+dev1d123@users.noreply.github.com>
Date: Sat May 18 22:19:26 2024 -0500

    Version Final

    Se agrega una tabla para almacenar los elementos, se añaden mas estilos y se implementan los signos raros como sqrt o pow.

commit 265a3a74ce978868b005588bbaf9e81ceac4ecaf
Author: dev1d123 <139636102+dev1d123@users.noreply.github.com>
Date: Sat May 18 22:03:08 2024 -0500

    Se completan los estilos para el formulario

commit f7cc6c32de02455474979b3a92aa5cfe31a3a81b
Author: dev1d123 <dhuamano@unsa.edu.pe>
Date: Sat May 18 22:00:17 2024 -0500

    Se completaron las demas etiquetas para asimilar un formulario

commit 99a59b34c17a5e8404e293d51864328f0013cdf6
Author: dev1d123 <dhuamano@unsa.edu.pe>
Date: Sat May 18 20:44:23 2024 -0500

    Cambios de clean code

commit 1a120b25eb6399c99b5c02db270004784c046c89
Author: dev1d123 <dhuamano@unsa.edu.pe>
Date: Sat May 18 22:19:26 2024 -0500
... skipping ...
commit 81de50f3068a713c9f712dae2f869ca123fb9629 (HEAD -> main, origin/main, origin/HEAD)
Author: dev1d123 <139636102+dev1d123@users.noreply.github.com>
Date: Sat May 18 22:19:26 2024 -0500

    Version Final

    Se agrega una tabla para almacenar los elementos, se añaden mas estilos y se implementan los signos raros como sqrt o pow.

commit 265a3a74ce978868b005588bbaf9e81ceac4ecaf
Author: dev1d123 <139636102+dev1d123@users.noreply.github.com>
Date: Sat May 18 22:03:08 2024 -0500

    Se completan los estilos para el formulario

commit f7cc6c32de02455474979b3a92aa5cfe31a3a81b
Author: dev1d123 <dhuamano@unsa.edu.pe>
Date: Sat May 18 22:00:17 2024 -0500

    Se completaron las demas etiquetas para asimilar un formulario
```

Figura 1: Imagen de los commits con git log

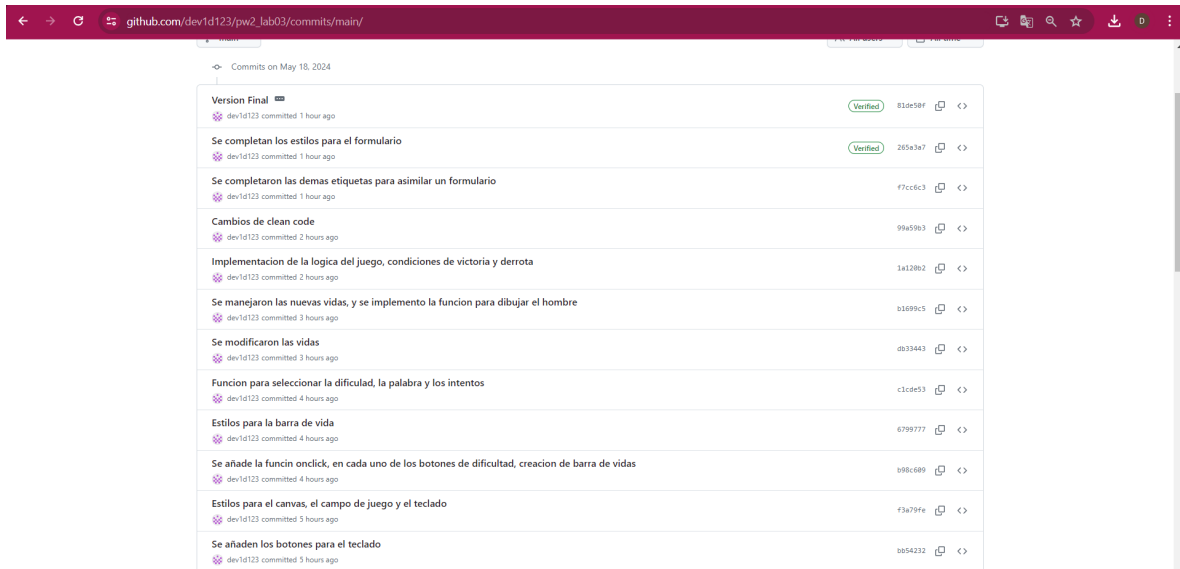


Figura 2: Imagen de los commits en el repositorio

4. Ejercicios propuestos

4.1. Ejercicio 1

- Documento HTML, se trato de imitar el formulario propuesto en la guia, a modo de practica, se añadireon los botones (button) con sus respectivas clases, debido a que se utilizara sus atributos text.content.

Listing 1: Ejercicio1/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulario Bancario</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Formulario Bancario</h1>
  </header>
  <div class="form-container">
    <form action="#" method="post">
      <div class="form-group">
        <label for="bank">Banco:</label>
        <select name="bank" id="bank">
          <option value="bvp">BPC</option>
          <option value="interbank">Interbank</option>
          <option value="scotiabank">Scotiabank</option>
        </select>
      </div>
      <div class="form-group">
```

```

        <label for="card-number">Numero de Tarjeta:</label>
        <input type="number" name="card-number" id="card-number" required>
    </div>
    <div class="form-group">
        <label>Tipo de DNI:</label>
        <label for="normal"><input type="radio" name="tipodni" id="normal"
            value="normal" checked> Normal</label>
        <label for="electronico"><input type="radio" name="tipodni" id="electronico"
            value="electronico"> Electronico</label>
    </div>
    <div class="form-group">
        <label for="dni-number">Numero de DNI:</label>
        <input type="number" name="dni-number" id="dni-number" required>
    </div>
    <div class="container">
        <input type="text" name="text" id="in" class="in" dir="rtl">
        <div class="buttons">
            <button type="button" class="b">1</button>
            <button type="button" class="b">2</button>
            <button type="button" class="b">3</button>
            <button type="button" class="b">4</button>
            <button type="button" class="b">5</button>
            <button type="button" class="b">6</button>
            <button type="button" class="b">7</button>
            <button type="button" class="b">8</button>
            <button type="button" class="b">9</button>
            <button type="button" class="b">0</button>
            <button type="button" id="clear" class="big">Clear</button>
        </div>
    </div>
    <div class="form-group">
        <a href="#" class="forgot-password">Olvide mi contrasena</a>
    </div>
    <div class="form-group">
        <button type="submit" class="submit">Enviar</button>
    </div>
</form>
</div>

<script src="script_ejercicio_01.js"></script>
</body>
</html>

```

- Archivo JavaScript, se utiliza un set para la generacion de los numeros aleatorios, mediante una iteracion, luego se copian estos datos a un arreglo, posteriormente se distribuyen los elementos de este arreglo a todos los botones mediante un querySelectorAll.

Listing 2: Ejercicio1/scriptejercicio01.js

```

console.log("Exito")
var elem = document.getElementsByTagName("body");
console.log(elem);
var random =Math.floor(Math.random()*10);
console.log("Numero aleatorio!", random);

```

```
var numerosAleatorios = new Set();

while(numerosAleatorios.size != 10){
    random =Math.floor(Math.random()*10);
    console.log("Numero aleatorio!", random);
    numerosAleatorios.add(random);
}
console.log(numerosAleatorios);
var arreglo = [];

numerosAleatorios.forEach(elem =>{
    arreglo.push(elem);
});

console.log(arreglo);

//obtener los elementos
var buttons = document.querySelectorAll(".b");
console.log(buttons);
var index = 0;
buttons.forEach(b =>{
    b.textContent = arreglo[index];
    index++;
});

//agregando funcionalidad a los botones
var input = document.getElementById("in");
console.log(input);
buttons.forEach(b =>{
    b.addEventListener('click', function(){
        console.log("click", b.textContent);
        input.value = input.value + b.textContent;
    });
});

var cl = document.getElementById("clear");
cl.addEventListener('click', function(){
    input.value = "";
});
```

- Imagen de la pagina.

Formulario Bancario

Banco:

Número de Tarjeta:

Tipo de DNI:
☒ Normal
☐ Electrónico

Número de DNI:

739

641

258

0Clear

[Olvidé mi contraseña](#)

Enviar

Figura 3: Ejecucion del Ejercicio 1

4.2. Ejercicio 2

- Documento HTML, se añadieron todos los botones que se muestran en la practica, para la pila se trato de imitar una usando una tabla

Listing 3: Ejercicio2/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculadora</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <p>Calculadora By David</p>
    <input type="text" name="text" id="in" class="in" dir="ltr">
    <div class="buttons">
      <button class="b">Del</button>
      <button class="b o">(</button>
      <button class="b o">)</button>
      <button class="b o">%</button>
      <button class="b">pi</button>
      <button class="b n">7</button>
      <button class="b n">8</button>
      <button class="b n">9</button>
      <button class="b o">/</button>
      <button class="b">sqrt</button>
      <button class="b n">4</button>
      <button class="b n">5</button>
      <button class="b n">6</button>
    </div>
  </div>
</body>
</html>
```

```
<button class="b o">*</button>
<button class="b">pow2</button>
<button class="b n">1</button>
<button class="b n">2</button>
<button class="b n">3</button>
<button class="b o">-</button>
<button id="calc" class="big">=</button>
<button class="b n">0</button>
<button class="b">.</button>
<button class="b o">%</button>
<button class="b o">+</button>
</div>
<div id="stack">
  <table id="stack-table">
    <thead>
      <tr>
        <th>Expresion</th>
        <th>Resultado</th>
      </tr>
    </thead>
    <tbody>
      <!-- Historial de calculos se inserta aca -->
    </tbody>
  </table>
</div>
</div>
<script src="script_ejercicio_02.js"></script>
</body>
</html>
```

- Archivo JavaScript, se divide en dos grandes partes: Ingresar los datos a la pantalla y procesar la pila.

Listing 4: Ejercicio2/scriptejercicio02.js

```
console.log("Conexion exitosa");
var input = document.getElementById("in");

var basic = document.querySelectorAll(".b");
basic.forEach(b => {
  b.addEventListener('click', function() {
    console.log("click", b.textContent);
    input = document.getElementById("in");
    if (b.textContent === "Del") {
      input.value = input.value.slice(0, -1);
    } else if (b.textContent === "pi") {
      input.value += Math.PI;
    } else if (b.textContent === "sqrt") {
      input.value += "Math.sqrt(";
    } else if (b.textContent === "pow2") {
      input.value += "**2";
    } else {
      input.value += b.textContent;
    }
  });
});
```

- Imagen de la pagina.



- Documento HTML, se divide en dos, una ventana para escoger la dificultad del juego, cuya logica sera implementada con "display:none", y el area de juego denotada por la clase "field", posee mensajes indicando varios parametros utiles como la palabra, el canva y al final se añade un elemento para mostrar las vidas, esto a modo de practica.

Página 8


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>El ahorcado!</title>
</head>
<body>
  <div id = "menu-inicial" class="menu-inicial">
    <p>Bienvenido al juego del ahorcado</p>
    <ol>
      <li>El juego comienza con una palabra secreta que debe ser adivinada.</li>
      <li>El jugador intenta adivinar la palabra proponiendo letras, una a la vez.</li>
      <li>Si la letra propuesta esta en la palabra secreta, se revela en su posicion correspondiente.</li>
      <li>Si la letra propuesta no esta en la palabra secreta, se agrega una parte del dibujo del ahorcado.</li>
      <li>El juego continua hasta que:
        <ul>
          <li>El jugador adivine todas las letras de la palabra (victoria), o</li>
          <li>El dibujo del ahorcado se complete (derrota).</li>
        </ul>
      </li>
    </ol>
    <div class="options">
      <div class="diff easy">
        <p>Tamano: 5</p>
        <p>Errores: 6</p>
        <button id="easy" onclick="diffSelection(1)">FACIL</button>
      </div>
      <div class="diff medium">
        <p>Tamano: 8</p>
        <p>Errores: 3</p>
        <button id="medium" onclick="diffSelection(2)">MEDIO</button>
      </div>
      <div class="diff hard">
        <p>Tamano: 12</p>
        <p>Errores: 2</p>
        <button id="hard" onclick="diffSelection(3)">DIFICIL</button>
      </div>
    </div>
  </div>

  <div class="field">
    <p class="logo">EL AHORCADO!</p>
    <p>Dificultad: <span id="diff"></span></p>
    <div class="game">
      <div class="draw">
        <canvas id="myCanvas" width="200" height="150" style="border:1px solid #000000;">
      </div>
      <p>Palabras adivinadas</p>
      <p class="word" id="word"></p>
    </div>
    <div class="teclas">
```

```

        <button class="kw">q</button>
        <button class="kw">w</button>
        <button class="kw">e</button>
        <button class="kw">r</button>
        <button class="kw">t</button>
        <button class="kw">y</button>
        <button class="kw">u</button>
        <button class="kw">i</button>
        <button class="kw">o</button>
        <button class="kw">p</button>
        <button class="kw">a</button>
        <button class="kw">s</button>
        <button class="kw">d</button>
        <button class="kw">f</button>
        <button class="kw">g</button>
        <button class="kw">h</button>
        <button class="kw">j</button>
        <button class="kw">k</button>
        <button class="kw">l</button>
        <!--No se puede representar la letra adecuada en tex-->
        <button class="kw">n</button>
        <button class="kw">z</button>
        <button class="kw">x</button>
        <button class="kw">c</button>
        <button class="kw">v</button>
        <button class="kw">b</button>
        <button class="kw">n</button>
        <button class="kw">m</button>
    </div>

</div>
<div class="container">
    <div class="lives">
        <p>Intentos restantes <span id="lives"></span></p>
    </div>
</div>

<script src="script_ejercicio_03.js"></script>
</body>
</html>

```

- Archivo JavaScript, al inicio se definen los arreglos de palabras para cada dificultad, luego se declaran variables globales debido a que estas se modificaran segun la dificultad.
- Comienza con la seleccion de dificultad, cada funcion tiene el objetivo de cambiar las variables de errores, elegir la palabra adecuada, etc. Aqui se usa la funcion onclick().
- Se realiza una funcion para graficar al monigote en funcion del estado del juego, se implementan intrincados metodos para buscar la forma, posicion y tamaño en canvas.
- Por ultimo se implementa la ya conocida modalidad del juego del ahorcado, se añaden todos las funciones necesarias para actualizar la interfaz.

Listing 6: Ejercicio3/scriptejercicio03.js

```
//Arreglos para las palabras!

const arrEasy = [
    "manos",
    "lapis",
    "playa",
    "campo",
    "salud",
    "fruta",
    "amigo",
    "carne",
    "huevo",
    "perro"
];

const arrMedium = [
    "elefante",
    "computar",
    "edificio",
    "hospital",
    "montanas",
    "mochilas",
    "ardillas",
    "delfines",
    "paraguas",
    "escobedo"
];

const arrHard = [
    "construccion",
    "constelacion",
    "refrigerador",
    "organizacion",
    "malentendido",
    "medioambiente"
];

var tries;
var word;
var elem;
var lives;
//obtener los botones
let menuInicial = document.getElementById("menu-inicial");
let messageDiff = document.getElementById("diff");
function diffSelection(cod){
    console.log("Se selecciono, ", cod);
    menuInicial.style.display = "none";
    if(cod == 1){
        messageDiff.innerHTML = "Facil";
        easyGame();
    }else if(cod == 2){
        messageDiff.innerHTML = "Medio";
        mediumGame();
    }else{
        messageDiff.innerHTML = "Dificil";
        hardGame();
    }
}
```

```
    }  
    game(tries, word, elem, lives);  
}  
  
function easyGame(){  
    tries = 6;  
    //elegir la palabra!  
    var index = Math.floor(Math.random()*11);  
    word = arrEasy[index];  
    console.log("La palabra es: ", word);  
    //definiendo los espacios en blanco!  
    elem = document.getElementById("word");  
    lives = document.getElementById("lives");  
    l = "";  
    s = "";  
    for(var i = 0; i < 5; i++){  
        s += "*";  
    }  
    for(var i = 0; i < tries; i++){  
        l += "&#128515"  
    }  
    elem.innerHTML = s;  
    lives.innerHTML = l;  
    window.scrollTo(0,50);  
}  
  
function mediumGame(){  
    tries = 3;  
    //elegir la palabra!  
    var index = Math.floor(Math.random()*11);  
    word = arrMedium[index];  
    console.log("La palabra es: ", word);  
    //definiendo los espacios en blanco!  
    elem = document.getElementById("word");  
    lives = document.getElementById("lives");  
    l = "";  
    s = "";  
    for(var i = 0; i < 8; i++){  
        s += "*";  
    }  
    for(var i = 0; i < tries; i++){  
        l += "&#128515"  
    }  
    elem.innerHTML = s;  
    lives.innerHTML = l;  
    window.scrollTo(0,50);  
}  
  
function hardGame(){  
    tries = 2;  
    //elegir la palabra!
```

```
index = Math.floor(Math.random()*7);
word = arrHard[index];
console.log("La palabra es: ", word);
//definiendo los espacios en blanco!
elem = document.getElementById("word");
lives = document.getElementById("lives");
l = "";
s = "";
for(var i = 0; i < 12; i++){
    s += "*";
}
for(var i = 0; i < tries; i++){
    l += "&#128515"
}
elem.innerHTML = s;
lives.innerHTML = l;
window.scrollTo(0,50);
}

//Se dibujara una fase de cada dibujo!
const canvas = document.getElementById('myCanvas');
const ctx = canvas.getContext('2d');

function drawHangman(stage) {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    ctx.strokeStyle = 'black';
    ctx.lineWidth = Math.max(2, Math.floor(canvas.width / 200));
    console.log(canvas.width)
    const scale = 0.40; //definir el tamaño del lienzo
    const startX = 20 * scale;
    const startY = canvas.height - 20 * scale;
    const headX = 200 * scale;
    const headY = 80 * scale;
    const bodyY = 110 * scale;
    const armY = 140 * scale;
    const armLeftX = 150 * scale;
    const armRightX = 250 * scale;
    const legY = 250 * scale;
    const legTopX = 200 * scale;
    const legLeftX = 150 * scale;
    const legRightX = 250 * scale;
    const legBottomY = 330 * scale;

    // Dibujar la horca
    ctx.beginPath();
    ctx.moveTo(startX, startY);
    ctx.lineTo(startX, 20 * scale);

    ctx.moveTo(startX - 5, startY);
    ctx.lineTo(startX - 5, 20 * scale);

    ctx.lineTo(200 * scale, 20 * scale);
```

```
ctx.lineTo(200 * scale, 50 * scale);
ctx.stroke();

//cabeza
if (stage >= 1) {
  ctx.beginPath();
  ctx.arc(headX, headY, 30 * scale, 0, Math.PI * 2);
  ctx.stroke();
}

//linea del cuerpo
if (stage >= 2) {
  ctx.beginPath();
  ctx.moveTo(headX, bodyY);
  ctx.lineTo(headX, 250 * scale);
  ctx.stroke();
}

//linea del brazo izq
if (stage >= 3) {
  ctx.beginPath();
  ctx.moveTo(headX, armY);
  ctx.lineTo(armLeftX, 200 * scale);
  ctx.stroke();
}

//linea del brazo derecho
if (stage >= 4) {
  ctx.beginPath();
  ctx.moveTo(headX, armY);
  ctx.lineTo(armRightX, 200 * scale);
  ctx.stroke();
}

//linea de la pierna izq
if (stage >= 5) {
  ctx.beginPath();
  ctx.moveTo(legTopX, legY);
  ctx.lineTo(legLeftX, legBottomY);
  ctx.stroke();
}

//linea de la pierna der
if (stage >= 6) {
  ctx.beginPath();
  ctx.moveTo(legTopX, legY);
  ctx.lineTo(legRightX, legBottomY);
  ctx.stroke();
}
}
drawHangman(0);

function game(intentos, palabra, texto, vidas) {
  let totalTries = intentos;

  console.log("Intentos: ", intentos);
```

```
console.log("palabra: ", palabra);
console.log("texto: ", texto);
console.log("vidas: ", vidas);

let progreso = texto.textContent.split('');

function actualizarProgreso(letra) {
  let encontrado = false;
  for (let i = 0; i < palabra.length; i++) {
    if (palabra[i] === letra) {
      progreso[i] = letra;
      encontrado = true;
    }
  }
  return encontrado;
}

function haGanado() {
  return progreso.join('') === palabra;
}

function actualizarPantalla() {
  texto.textContent = progreso.join('');
  if(totalTries == 6){
    drawHangman(totalTries-intentos);
  }else if(totalTries == 3){
    drawHangman((totalTries-intentos) * 2);
  }else{
    drawHangman((totalTries-intentos) * 3);
  }
}

function manejarLetra(letra) {
  if (!actualizarProgreso(letra)) {
    intentos--;
    vidas.textContent = "";
    var l = "";
    for(var i = 0; i < intentos; i++){
      l += "&#128515"
    }
    vidas.innerHTML = l;
  }
  if (haGanado()) {
    alert("Has ganado La palabra era: " + palabra);
    location.reload();

  } else if (intentos === 0) {
    alert("Has perdido La palabra era: " + palabra);
    location.reload();
  }

  actualizarPantalla();
}

document.querySelectorAll('.kw').forEach(button => {
```

```
button.addEventListener('click', () => {
    let letra = button.textContent;
    manejarLetra(letra);
});

actualizarPantalla();
}
```

- Imagen de la pagina.



Figura 5: Ejecucion del Ejercicio 3

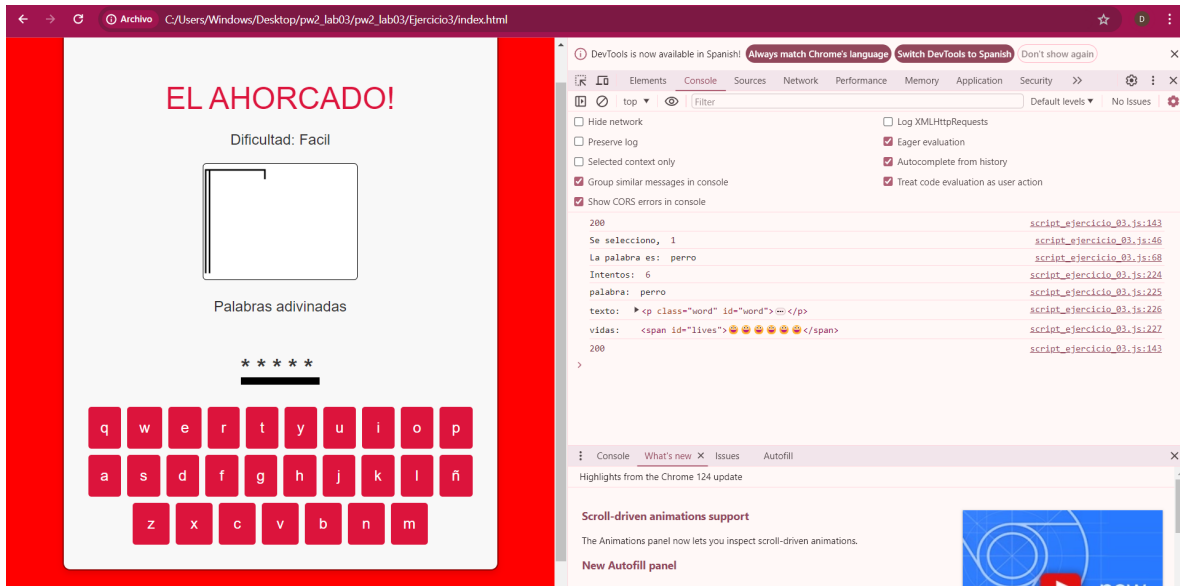


Figura 6: Ejecucion del Ejercicio 3

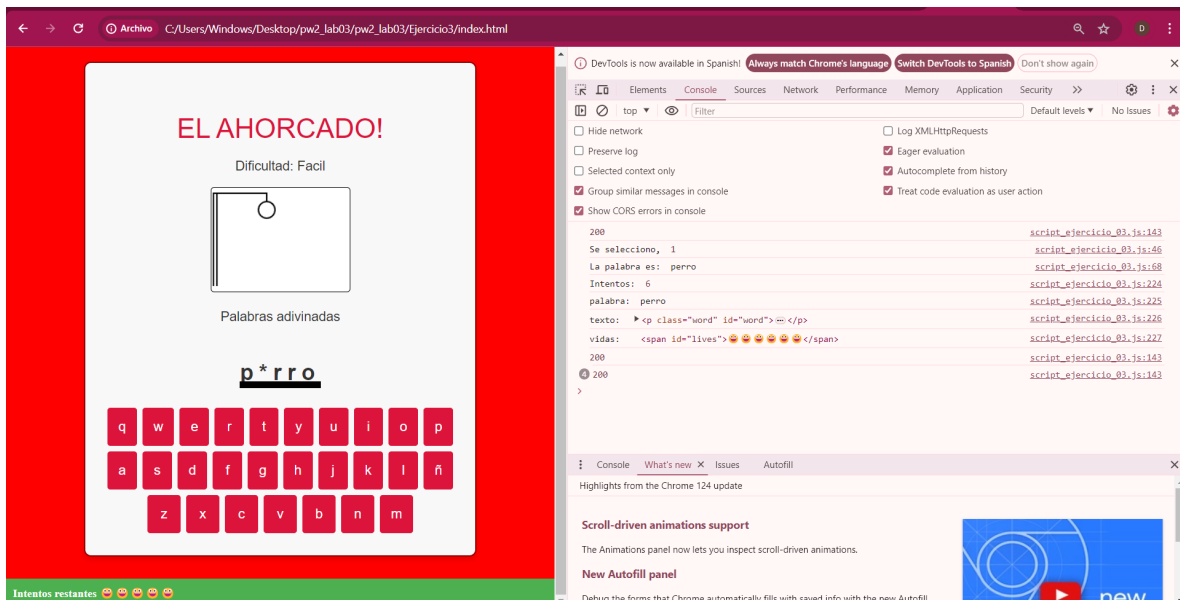


Figura 7: Ejecucion del Ejercicio 3

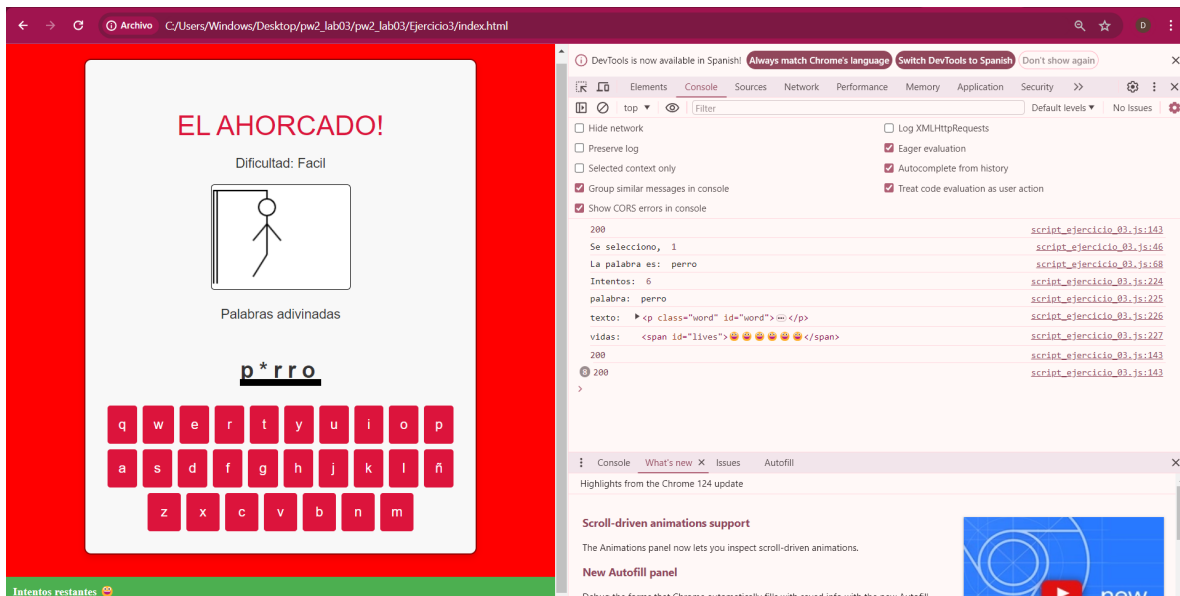


Figura 8: Ejecucion del Ejercicio 3

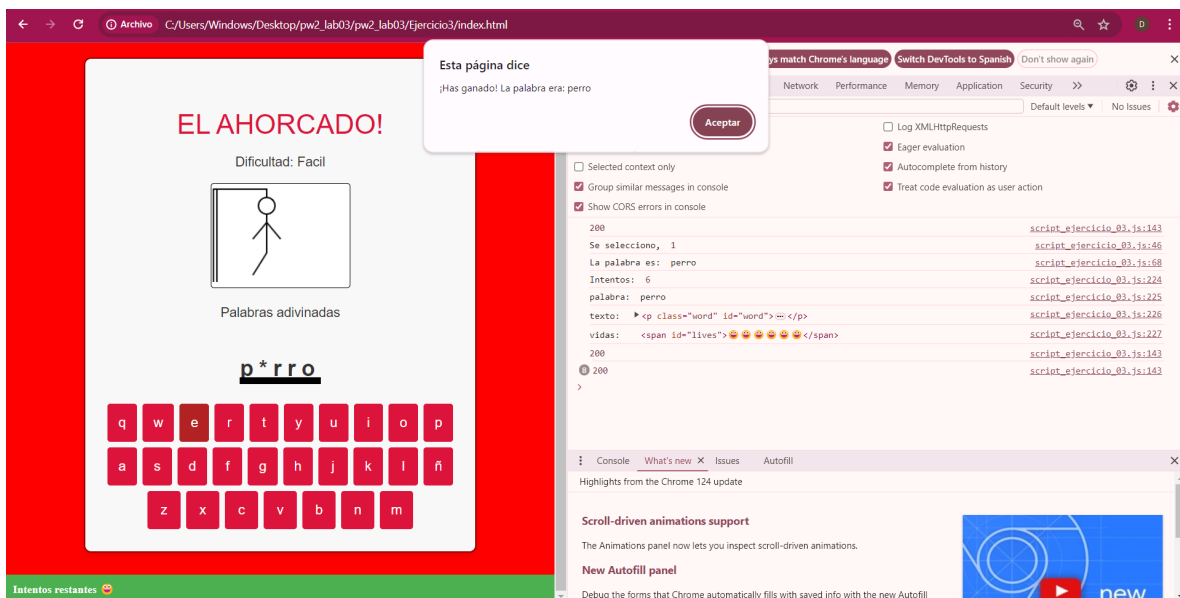


Figura 9: Ejecucion del Ejercicio 3

5. Pregunta

5.1. Explique una herramienta para ofuzcar código JavaScript.

- JavaScript Obfuscator es una herramienta ampliamente utilizada para ofuscar código JavaScript. Esta herramienta transforma el código legible y entendible en una versión que es difícil de leer y comprender para los humanos, pero que sigue siendo funcional para los navegadores y entornos de ejecución.

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

- Mostrando código ofuscado del ejercicio 2

```
var _0xcd3575=_0x5b47;function _0x5b47(_0x111c8a,_0x5a4834){var _0x4c0086=_0x4c00();return _0x5b47=function(_0x5b47e8,_0x14a587){_0x5b47e8=_0x5b47e8-0x1a3;var _0x8b8d36=_0x4c0086[_0x5b47e8];return _0x8b8d36;},_0x5b47(_0x111c8a,_0x5a4834);}function _0x4c00(){var _0x1da752=['148524AHJlox','insertRow','Conexion\x20exitosa','170266NFmbsz','querySelector','querySelectorAll','sqrt','Error','\x20expresion\x20incorrecta','click','addEventListener','getElementById','**2','138015KuEYyv','3040429DKyRef','log','insertCell','pow2','431428tUqJvJ','#stack-table\x20tbody','3CbzRKR','8UTbBrK','value','1187605RaXHEu','textContent','10vyADhw','forEach','110GgSgrp','calc','301920PHztIv'];_0x4c00=function(){return _0x1da752;};return _0x4c00();}(function(_0x1e6302,_0x48ed23){var _0x2384c1=_0x5b47,_0x375f8e=_0x1e6302();while(![]){try{var _0x2de794=-parseInt(_0x2384c1(0x1bb))/0x1+-parseInt(_0x2384c1(0x1b7))/0x2+-parseInt(_0x2384c1(0x1ae))/0x3*(-parseInt(_0x2384c1(0x1ac))/0x4)-parseInt(_0x2384c1(0x1b3))/0x5*(-parseInt(_0x2384c1(0x1b8))/0x6)+parseInt(_0x2384c1(0x1a8))/0x7*(-parseInt(_0x2384c1(0x1af))/0x8)+parseInt(_0x2384c1(0x1a7))/0x9*(-parseInt(_0x2384c1(0x1b5))/0xa)+parseInt(_0x2384c1(0x1b1))/0xb;if(_0x2de794===_0x48ed23)break;else _0x375f8e['push'](_0x375f8e['shift']());}catch(_0x115def){_0x375f8e['push'](_0x375f8e['shift']());}}(_0x4c00,0x3d132),console[_0xcd3575(0x1a9)](_0xcd3575(0x1ba));var input=document[_0xcd3575(0x1a5)]('in'),basic=document[_0xcd3575(0x1bd)]('b');basic[_0xcd3575(0x1b4)](_0x5cf563=>{var _0x17a526=_0xcd3575;_0x5cf563[_0x17a526(0x1a4)](_0x17a526(0x1a3),function(){var _0x29a39e=_0x17a526;console['log'](_0x29a39e(0x1a3),_0x5cf563['textContent']),input=document['getElementById']('in');if(_0x5cf563['textContent']===_0x29a39e(0x1b0))input[_0x29a39e(0x1b0)]=input[_0x29a39e(0x1b0)]['slice'](0x0,-0x1);else if(_0x5cf563['textContent']===_0x29a39e(0x1be))input[_0x29a39e(0x1b0)]=_0x29a39e(0x1b0)+Math['PI'];else if(_0x5cf563['textContent']===_0x29a39e(0x1b2))input[_0x29a39e(0x1b0)]=_0x29a39e(0x1a6):input[_0x29a39e(0x1b0)]=_0x5cf563[_0x29a39e(0x1b2)];});});var calcular=document[_0xcd3575(0x1a5)](_0xcd3575(0x1b6)),stackTableBody=document[_0xcd3575(0x1bc)](_0xcd3575(0x1ad));calcular[_0xcd3575(0x1a4)](_0xcd3575(0x1a3),function(){var _0x937615=_0xcd3575;try{var _0x3b8f31=input[_0x937615(0x1b0)],_0x27ae36=eval(_0x3b8f31);input[_0x937615(0x1b0)]=_0x27ae36;var _0x208257=stackTableBody[_0x937615(0x1b9)](0x0),_0x4a7816=_0x208257['insertCell'](0x0),_0x41a981=_0x208257[_0x937615(0x1aa)](0x1);_0x4a7816[_0x937615(0x1b2)]=_0x3b8f31,_0x41a981[_0x937615(0x1b2)]=_0x27ae36,input['value']='';}catch(_0x5a3cc0){alert(_0x937615(0x1bf)),input[_0x937615(0x1b2)]=_0x27ae36;}});}
```

Figura 10: Código js del Ejercicio 2 ofuscado

5.3. Adjunte a su repositorio ambas versiones:

- Se implementaron los ejemplos en el repositorio

6. Rúbricas

6.1. 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 fácil de leer.

6.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple 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	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	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	2	
Total		20		18	

7. Referencias

- <https://www.w3schools.com/js/default.asp>