

## Laboratorio 02

### Tema: JavaScript

Nota

Estudiante	Escuela	Asignatura
Kevin Gonzales Fernandez Kgonzalesf@ulasalle.edu.pe	Carrera Profesional de Ingeniería de Software	Ingeniería Web Semestre: VIII Código: 3.8.6.16

Laboratorio	Tema	Duración
02	JavaScript	06 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	28 Agosto 2023	02 Setiembre 2023

## 1. Tarea

### 1.1. Ejercicio 01

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

#### 1.1.1. Código HTML

En el código Html se mostrara la estructura de la página web, ubicando y cargando los códigos Css junto al javascript, para luego organizar los lugares y funciones donde estará ubicado cada cosa del juego del ahorcado.

Listing 1: Code HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>El Juego Interactivo del Ahorcado</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="header">
    <h1>El juego interactivo del ahorcado</h1>
  </div>
```

```
<div class="container">
  <canvas id="canvas" width="400" height="400"></canvas>
  <button id="playButton">Presionar</button>
</div>
<div id="winModal" class="modal">
  <div class="modal-content">
    <span class="close">&times;</span>
    <p>Felicidades, has ganado!</p>
    <button id="restartButton">Comenzar de Nuevo</button>
  </div>
</div>
<script src="script.js"></script>
</body>
</html>
```

### 1.1.2. Código CSS

En el código CSS se nos muestra los colores y diseños que se usará en la página web del juego del ahorcado, dándole colores a los botones y al fondo de esta misma.

Listing 2: Code CSS

```
body {
  margin: 0;
  padding: 0;
  background-color: lightgreen;
}

.header {
  background-color: darkgreen;
  text-align: center;
  padding: 20px 0;
}

h1 {
  color: #fbff00;
  font-size: 20px;
}

.container {
  text-align: center;
}

#canvas {
  background-color: lightgreen;
  border: 2px solid darkgreen;
  margin-top: 20px;
}

#playButton {
  position: absolute; /* Aaido para posicionar el botn */
  bottom: 10px; /* Ajuste para posicionar debajo de la imagen */
  left: 50%; /* Aaido para centrar horizontalmente */
  transform: translateX(-50%); /* Aaido para centrar horizontalmente */
  padding: 10px 20px;
  font-size: 16px;
}
```

```
background-color: red;
color: white;
border: none;
cursor: pointer;
}

.modal {
  display: none;
  position: fixed;
  z-index: 1;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0,0,0,0.5);
}

.modal-content {
  background-color: white;
  margin: 25% auto;
  padding: 20px;
  border: 1px solid #888;
  width: 50%;
}

.close {
  color: #aaa;
  float: right;
  font-size: 28px;
  font-weight: bold;
  cursor: pointer;
}

.close:hover,
.close:focus {
  color: black;
  text-decoration: none;
  cursor: pointer;
}

#restartButton {
  padding: 10px 20px;
  font-size: 16px;
  background-color: orange;
  color: #ffffff;
  border: none;
  cursor: pointer;
}
```

### 1.1.3. Código Javascript

En el código javascript se mostrara todas las mecánicas y funciones que darán vida al juego del ahorcado, como la de presionar los botones, el canvas y la ventana extra.

Listing 3: Code HTML

```
document.addEventListener("DOMContentLoaded", function() {
```

```
const canvas = document.getElementById('canvas');
const ctx = canvas.getContext('2d');
const playButton = document.getElementById('playButton');
const winModal = document.getElementById('winModal');
const restartButton = document.getElementById('restartButton');
const closeButton = document.querySelector('.close');

const drawHanger = () => {
  ctx.beginPath();
  ctx.moveTo(50, 350);
  ctx.lineTo(150, 350);
  ctx.lineTo(100, 300);
  ctx.lineTo(50, 350);
  ctx.lineTo(50, 50);
  ctx.lineTo(200, 50);
  ctx.lineTo(200, 100);
  ctx.stroke();
};

const drawMan = (step) => {
  switch (step) {
    case 1:
      ctx.beginPath();
      ctx.arc(200, 150, 50, 0, Math.PI * 2);
      ctx.stroke();
      break;
    case 2:
      ctx.beginPath();
      ctx.moveTo(200, 200);
      ctx.lineTo(200, 300);
      ctx.stroke();
      break;
    case 3:
      ctx.beginPath();
      ctx.moveTo(200, 220);
      ctx.lineTo(150, 250);
      ctx.stroke();
      break;
    case 4:
      ctx.beginPath();
      ctx.moveTo(200, 220);
      ctx.lineTo(250, 250);
      ctx.stroke();
      break;
    case 5:
      ctx.beginPath();
      ctx.moveTo(200, 300);
      ctx.lineTo(150, 350);
      ctx.stroke();
      break;
    case 6:
      ctx.beginPath();
      ctx.moveTo(200, 300);
      ctx.lineTo(250, 350);
      ctx.stroke();
      break;
  }
};
```

```
        default:
            break;
    }
};

const drawGameOver = () => {
    winModal.style.display = 'block';
};

const clearCanvas = () => {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    drawHanger();
};

let gameStarted = false;
let gameOver = false;
let step = 0;

playButton.addEventListener('click', () => {
    if (!gameOver) {
        if (!gameStarted) {
            clearCanvas();
            gameStarted = true;
        } else {
            step++;
            drawMan(step);
            if (step >= 6) {
                drawGameOver();
                gameOver = true;
            }
        }
    }
});

restartButton.addEventListener('click', () => {
    winModal.style.display = 'none';
    gameStarted = false;
    gameOver = false;
    step = 0;
    clearCanvas();
});

closeButton.addEventListener('click', () => {
    winModal.style.display = 'none';
});
});
```

#### 1.1.4. Ejecución de la página web

Procederemos a mostrar como funciona la pagina web del ahorcado, para jugar tendrás que presionar el botón rojo, el cual hará que se vaya construyendo de poco en poco el ahorcado, para cuando lo completes te saldrá una ventana de felicidades.

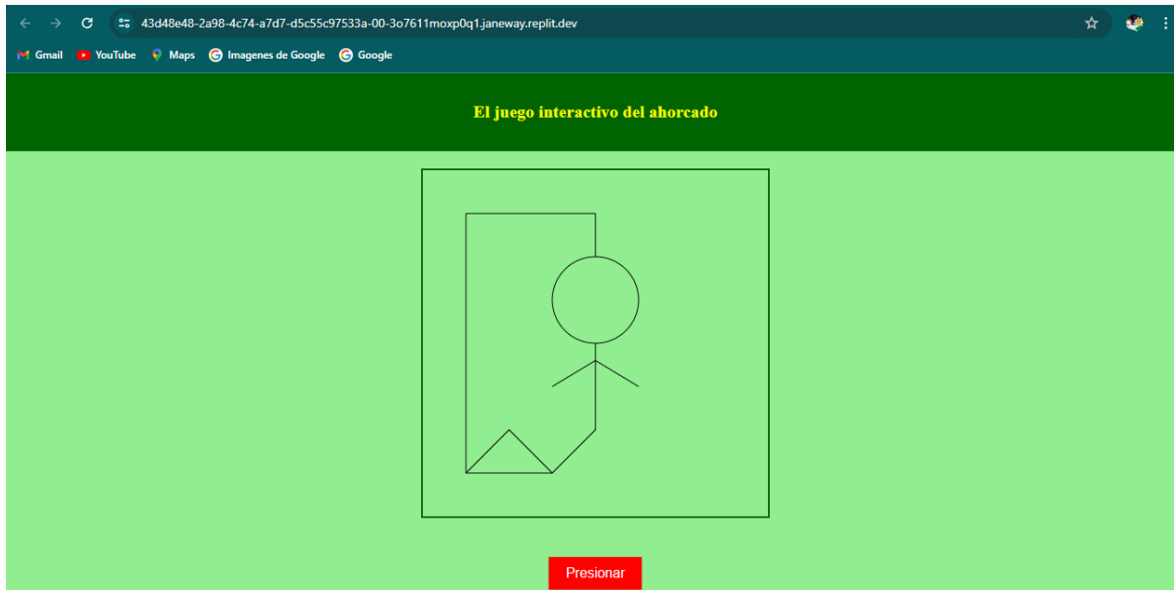


Figura 1: Ejecución del juego Ahorcado

## 1.2. 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.

### 1.2.1. Código HTML

En esta parte del código mostraremos la pagina web en el navegador, para ello el código recoge los códigos Css y javascript para mostrar los colores y funciones de la calculadora.

Listing 4: Code HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculadora con Pila</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="barra"></div>
  <div class="calculadora">
    <input type="text" id="display" readonly>
    <div class="botones">
      <button onclick="agregarCaracter('(')"></button>
      <button onclick="agregarCaracter(')')"></button>
```

```
<button onclick="limpiarDisplay()">C</button>
<button onclick="borrarCaracter()">Borrar</button>
<button onclick="agregarCaracter('pi')"></button>
<button onclick="agregarCaracter('Math.sqrt()')"></button>
<button onclick="agregarCaracter('%')">%</button>
<button onclick="agregarCaracter('n')">n</button>
<button onclick="agregarCaracter('1')">1</button>
<button onclick="agregarCaracter('2')">2</button>
<button onclick="agregarCaracter('3')">3</button>
<button onclick="agregarCaracter('+')">+</button>
<button onclick="agregarCaracter('4')">4</button>
<button onclick="agregarCaracter('5')">5</button>
<button onclick="agregarCaracter('6')">6</button>
<button onclick="agregarCaracter('-')">-</button>
<button onclick="agregarCaracter('7')">7</button>
<button onclick="agregarCaracter('8')">8</button>
<button onclick="agregarCaracter('9')">9</button>
<button onclick="agregarCaracter('*')">*</button>
<button onclick="agregarCaracter('0')">0</button>
<button onclick="agregarCaracter('.')">.</button>
<button onclick="calcular()">=</button>
<button onclick="agregarCaracter('/')">/</button>
<button onclick="agregarCaracter('*2')">X</button>
<button onclick="agregarCaracter('*')">x</button>
<button onclick="agregarCaracter('Math.E')">e</button>
</div>
</div>
<div class="pilas">
  <div class="lista-pilas" id="lista-pilas">
    <!-- Aqu se mostrarn las pilas -->
  </div>
  <button onclick="borrarPila()">Borrar Pila</button>
  <button onclick="editarPila()">Editar Pila</button>
</div>
<div id="modal" class="modal">
  <div class="modal-content">
    <span class="close" onclick="cerrarModal()">&times;</span>
    <p id="modal-title"></p>
    <textarea id="modal-input"></textarea>
    <button onclick="confirmarModal()">Confirmar</button>
  </div>
</div>
<script src="script.js"></script>
</body>
</html>
```

Código CSS En esta parte del código mostraremos los diseños, colores y botones que se muestra en la calculadora, también regulando su tamaño y ancho.

Listing 5: Code CSS

```
body {
margin: 0;
padding: 0;
background-color: #eaf2e3; /* Verde claro */
}
```

```
.barra {
  width: 100%;
  height: 50px;
  background-color: #2e7d32; /* Verde oscuro */
}

.calculadora {
  margin: 20px auto;
  width: 300px;
  background-color: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}

#display {
  width: calc(100% - 40px);
  height: 40px;
  margin-bottom: 10px;
  font-size: 20px;
  padding: 5px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

.botones {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 5px;
}

button {
  width: calc(100% - 10px);
  height: 40px;
  font-size: 18px;
  border: none;
  border-radius: 5px;
  background-color: #2e7d32;
  color: #fff;
  cursor: pointer;
}

button:hover {
  background-color: #5cb85c;
}

.pilas {
  margin-top: 20px;
  padding: 20px;
  background-color: #fff;
  border-radius: 10px;
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}

.lista-pilas {
  max-height: 200px;
```



```
        overflow-y: auto;
        border: 1px solid #ccc;
        border-radius: 5px;
        padding: 10px;
        margin-bottom: 10px;
    }

    .lista-pilas p {
        margin: 5px 0;
    }

    .modal {
        display: none;
        position: fixed;
        z-index: 1;
        left: 0;
        top: 0;
        width: 100%;
        height: 100%;
        background-color: rgba(0, 0, 0, 0.5);
    }

    .modal-content {
        background-color: #fefefe;
        margin: 20% auto;
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.5);
    }

    .close {
        color: #aaa;
        float: right;
        font-size: 28px;
        font-weight: bold;
    }

    .close:hover,
    .close:focus {
        color: black;
        text-decoration: none;
        cursor: pointer;
    }

    #modal-input {
        width: calc(100% - 20px);
        height: 100px;
        margin-top: 10px;
        padding: 10px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }
```

### 1.2.2. Código Javascript

En esta parte del código se mostrará todas las mecánicas y funciones que se pusieron, como la lógica de la calculadora como los botones y las pilas.

Listing 6: Code Javascript

```
let display = document.getElementById('display');
let listaPilas = document.getElementById('lista-pilas');
let operaciones = [];

function agregarCaracter(caracter) {
    display.value += caracter;
}

function borrarCaracter() {
    display.value = display.value.slice(0, -1);
}

function limpiarDisplay() {
    display.value = '';
}

function calcular() {
    try {
        let resultado = eval(display.value);
        operaciones.push({ operacion: display.value, resultado: resultado });
        mostrarPilas();
        display.value = resultado;
    } catch (error) {
        display.value = 'Error';
    }
}

function mostrarPilas() {
    listaPilas.innerHTML = '';
    for (let i = 0; i < operaciones.length; i++) {
        let p = document.createElement('p');
        p.textContent = operaciones[i].operacion + ' = ' + operaciones[i].resultado;
        listaPilas.appendChild(p);
    }
}

function borrarPila() {
    let indicePila = prompt('Ingrese el ndice de la pila a borrar (comenzando desde 1:');
    if (indicePila && !isNaN(indicePila) && indicePila > 0 && indicePila <=
        operaciones.length) {
        operaciones.splice(indicePila - 1, operaciones.length - indicePila + 1);
        mostrarPilas();
    } else {
        alert('ndice de pila invlido');
    }
}

function editarPila() {
    let indicePila = prompt('Ingrese el ndice de la pila a editar (comenzando desde 1:');
    if (indicePila && !isNaN(indicePila) && indicePila > 0 && indicePila <=
```

```
        operaciones.length) {
            mostrarModal('Editar Pila', 'Ingrese la nueva operacin:', (nuevaOperacion) => {
                operaciones.splice(indicePila - 1, operaciones.length - indicePila + 1);
                calcularNuevaOperacion(nuevaOperacion);
            });
        } else {
            alert('ndice de pila invlido');
        }
    }

function calcularNuevaOperacion(nuevaOperacion) {
    try {
        let resultado = eval(nuevaOperacion);
        operaciones.push({ operacion: nuevaOperacion, resultado: resultado });
        mostrarPilas();
        display.value = resultado;
    } catch (error) {
        display.value = 'Error';
    }
}

function mostrarModal(titulo, placeholder, callback) {
    let modal = document.getElementById('modal');
    let modalTitle = document.getElementById('modal-title');
    let modalInput = document.getElementById('modal-input');

    modal.style.display = 'block';
    modalTitle.textContent = titulo;
    modalInput.placeholder = placeholder;

    modalInput.value = '';
    modalInput.focus();

    let confirmarModal = document.querySelector('.modal-content button');
    confirmarModal.onclick = function () {
        modal.style.display = 'none';
        callback(modalInput.value);
    };
}

function cerrarModal() {
    let modal = document.getElementById('modal');
    modal.style.display = 'none';
}
```

### 1.2.3. Ejecución de la página web

Procederemos a mostrar el funcionamiento de la pagina web, al iniciar, se nos mostrara una calculadora donde podras realizar cálculos matemáticos, estos se iran guardando en pilas, las cuales puedes editar y borra según tu agrado

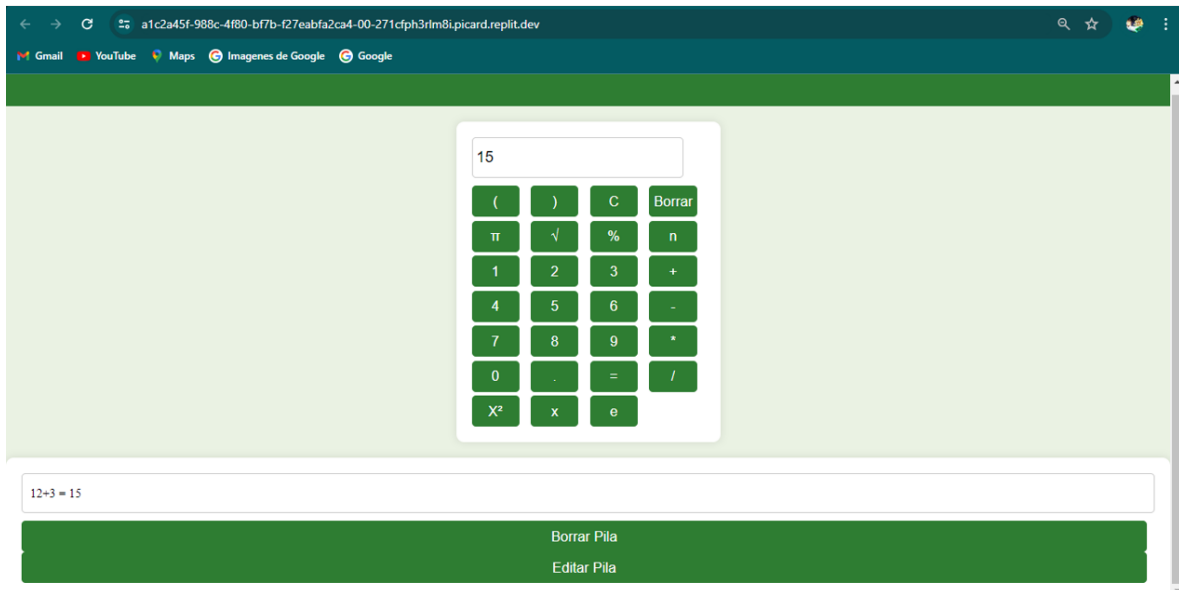


Figura 2: Ejecución de la calculadora

### 1.3. Ejercicio 03

Cree un teclado random para banca por internet

#### 1.3.1. Código HTML

En esta parte del código mostraremos la página web, vinculando los códigos CSS de diseño y JavaScript de funcionamiento, para que la hora de mostrarse la página web aparezca todo ordenado y con las condiciones que le puse.

Listing 7: Code HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pgina Web</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="header">
    
    
  </div>
  <div class="container">
    <div class="content">
      <div class="form">
        <p class="security-text">Se Encuentra en una Zona Segura</p>
        <label for="cardType">Selecccion:</label>
        <select name="cardType" id="cardType">
          <option value="Multired global debito">Multired global debito</option>
          <option value="Electronico debito">Electronico debito</option>
        </select>
      </div>
    </div>
  </div>
</body>
</html>
```

```
<option value="Monedero debito">Monedero debito</option>
<option value="debito">debito</option>
<option value="otros">otros</option>
</select>
<label for="cardNumber">Numero de targeta:</label>
<input type="text" id="cardNumber" maxlength="16">
<label for="documentType">Tipo de documento:</label>
<select name="documentType" id="documentType">
  <option value="DNI">DNI</option>
  <option value="Pasaporte">Pasaporte</option>
</select>
<input type="text" id="documentNumber" maxlength="12" placeholder="Nmero de
  documento">
<label for="internetKey">Generar tu clave por internet (solo hasta 6
  dgitos):</label>
<input type="password" id="internetKey" maxlength="6" placeholder="Ingresa tu
  clave de internet" readonly>
<div class="virtual-keyboard">
  <!-- Teclado virtual -->
</div>
<label for="captcha">Ingresar captcha:</label>
<div class="captcha-row">
  <div class="captcha">
    <!-- Captcha -->
  </div>
  <button id="captchaRefreshButton">Refrescar</button>
</div>
<input type="text" id="captchaInput" placeholder="Ingresa el captcha">
<button id="submitButton">Ingresar</button>
</div>
</div>
</div>
<div id="successModal" class="modal">
  <div class="modal-content">
    <span class="close">&times;</span>
    <p>Tus datos fueron registrados. Gracias por compartir tu informacin.</p>
    <button id="reloadButton">Volver a cargar datos</button>
  </div>
</div>
<script src="script.js"></script>
</body>
</html>
```

### 1.3.2. Código CSS

En esta parte del código se muestra los diseños, colores y matices que se usaron para darle color y forma a la página web, controlando estrictamente su tamaño y posición.

Listing 8: Code CSS

```
body {
margin: 0;
padding: 0;
background-color: lightblue;
font-family: Arial, sans-serif;
text-transform: capitalize;
```

```
}

.header {
  background-color: green;
  padding: 10px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.logo-left {
  width: 100px;
  height: auto;
}

.logo-right {
  width: 100px;
  height: auto;
}

.container {
  max-width: 600px;
  margin: 20px auto;
  background-color: white;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.content {
  display: flex;
  justify-content: center;
  align-items: center;
}

.form {
  margin-left: 20px;
}

.security-text {
  color: red;
}

label {
  display: block;
  margin-bottom: 5px;
}

input[type="text"],
select,
input[type="password"] {
  width: calc(100% - 20px);
  padding: 8px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
```

```
}

input[type="text"]:focus,
select:focus,
input[type="password"]:focus {
    outline: none;
    border-color: blue;
}

.virtual-keyboard {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 5px;
    margin-bottom: 10px;
}

.virtual-keyboard button {
    width: 100%;
    height: 40px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background-color: #fff;
    cursor: pointer;
}

.captcha-row {
    display: flex;
    align-items: center;
    margin-bottom: 10px;
}

.captcha {
    flex: 1;
    display: flex;
    align-items: center;
}

.captcha img {
    margin-right: 10px;
}

#captchaRefreshButton {
    margin-left: 10px;
}

#submitButton {
    background-color: red;
    color: white;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    cursor: pointer;
}

.modal {
    display: none;
}
```

```
position: fixed;
z-index: 1;
left: 50%;
top: 50%;
transform: translate(-50%, -50%);
width: 50%;
max-width: 400px;
background-color: rgba(0, 0, 0, 0.5);
text-align: center;
padding: 20px;
}

.modal-content {
background-color: white;
padding: 20px;
border-radius: 10px;
}

.close {
color: #aaa;
float: right;
font-size: 28px;
font-weight: bold;
cursor: pointer;
}

.close:hover,
.close:focus {
color: black;
text-decoration: none;
cursor: pointer;
}

#reloadButton {
background-color: red;
color: white;
border: none;
padding: 10px 20px;
border-radius: 5px;
cursor: pointer;
}

#reloadButton:hover {
background-color: darkred;
}
```

### 1.3.3. Código Javascript

En esta parte del código se nos mostrara todas las funcionalidades y mecánicas que tiene nuestra pagina web, como los botones, selección y escritura, ademas de guardar los datos.

Listing 9: Code Javascript

```
document.addEventListener("DOMContentLoaded", function() {
const captchaContainer = document.querySelector('.captcha');
const captchaInput = document.getElementById('captchaInput');
```



```
const captchaRefreshButton = document.getElementById('captchaRefreshButton');
const submitButton = document.getElementById('submitButton');
const successModal = document.getElementById('successModal');
const reloadButton = document.getElementById('reloadButton');
const cardTypeSelect = document.getElementById('cardType');
const cardNumberInput = document.getElementById('cardNumber');
const documentTypeSelect = document.getElementById('documentType');
const documentNumberInput = document.getElementById('documentNumber');
const internetKeyInput = document.getElementById('internetKey');
const virtualKeyboard = document.querySelector('.virtual-keyboard');

const generateCaptcha = () => {
  const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
  let captcha = '';
  for (let i = 0; i < 6; i++) {
    captcha += characters.charAt(Math.floor(Math.random() * characters.length));
  }
  return captcha;
};

let currentCaptcha = generateCaptcha();
captchaContainer.innerHTML = `${currentCaptcha}`;

captchaRefreshButton.addEventListener('click', () => {
  currentCaptcha = generateCaptcha();
  captchaContainer.innerHTML = `${currentCaptcha}`;
});

const handleSubmit = () => {
  successModal.style.display = 'block';
};

submitButton.addEventListener('click', handleSubmit);

reloadButton.addEventListener('click', () => {
  successModal.style.display = 'none';
  cardNumberInput.value = '';
  documentNumberInput.value = '';
  internetKeyInput.value = '';
  generateCaptchaImage();
});

documentTypeSelect.addEventListener('change', () => {
  const selectedOption = documentTypeSelect.value;
  if (selectedOption === 'DNI') {
    documentNumberInput.placeholder = 'Número de DNI (8 dígitos)';
    documentNumberInput.maxLength = 8;
  } else if (selectedOption === 'Pasaporte') {
    documentNumberInput.placeholder = 'Número de Pasaporte (12 dígitos)';
    documentNumberInput.maxLength = 12;
  }
});

const keyboardButtons = Array.from(virtualKeyboard.querySelectorAll('button'));
```

```
keyboardButtons.forEach(button => {
  button.addEventListener('click', () => {
    if (button.textContent === 'Limpiar') {
      internetKeyInput.value = '';
    } else {
      internetKeyInput.value += button.textContent;
    }
  });
});

const generateVirtualKeyboard = () => {
  const numbers = Array.from({ length: 9 }, (_, i) => i + 1);
  numbers.push(0);
  shuffleArray(numbers);
  const cleanButton = document.createElement('button');
  cleanButton.textContent = 'Limpiar';
  cleanButton.addEventListener('click', () => {
    internetKeyInput.value = '';
  });
  virtualKeyboard.innerHTML = '';
  numbers.forEach(number => {
    const button = document.createElement('button');
    button.textContent = number;
    button.addEventListener('click', () => {
      internetKeyInput.value += number;
    });
    virtualKeyboard.appendChild(button);
  });
  virtualKeyboard.appendChild(cleanButton); // Agregar el botn "Limpiar" al final
};

generateVirtualKeyboard();

const generateCaptchaImage = () => {
  currentCaptcha = generateCaptcha();
  captchaContainer.innerHTML = `${currentCaptcha}`;
};

captchaRefreshButton.addEventListener('click', generateCaptchaImage);

generateCaptchaImage();
});

function shuffleArray(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]];
  }
}
```

### 1.3.4. Ejecución de la página web

Procederemos a mostrar la pagina web bancaria, donde nos pedirá ingresar algunos datos y seleccionar algunas opciones, ademas el teclado virtual, algo importantes es que el teclado puede variar sus botones a distintas direcciones, por ultimo el captcha es aleatorio.

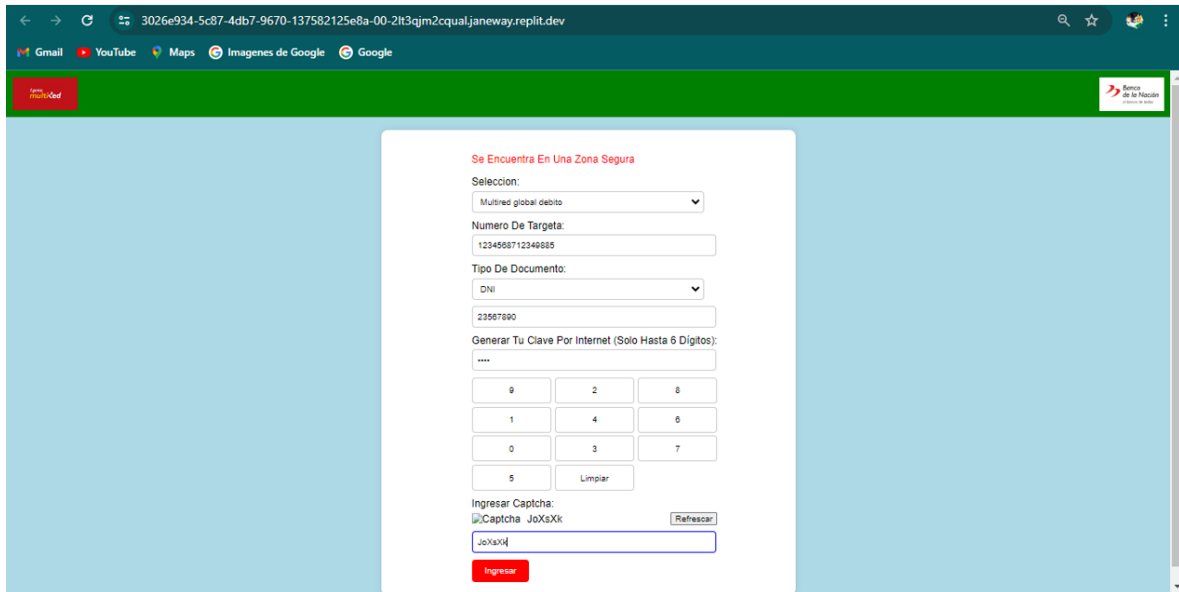


Figura 3: Ejecución de la banca web

## 1.4. Pregunta

a) Explique una herramienta para ofuzcar código JavaScript UglifyJS es una herramienta de código abierto que se utiliza para minificar y ofuscar código JavaScript. La minificación es el proceso de eliminar espacios en blanco, comentarios y otros caracteres no esenciales para reducir el tamaño del archivo. La ofuscación, por otro lado, es el proceso de modificar el código para que sea más difícil de entender para los humanos, pero que siga siendo funcional para las computadoras.

b) Muestre un ejemplo de su uso en uno de los ejercicios de la tarea. Procederemos a mostrar un ejemplo del código ofuscado con Banca de internet.js

```
document.addEventListener("DOMContentLoaded",function(){const
    e=document.querySelector(".captcha"),t=document.getElementById("captchaInput"),n=document.getElementById("captchaInput")
    e="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";let
    t="";for(let n=0;n
```

## 2. Repositorios

### 2.1. Development

<https://github.com/kevin1go-tech/CodePlusKevin.git>

### 2.2. Production

<https://github.com/kevin1go-tech/CodePlusKevin.git>