

## Informe de Laboratorio 03

### Tema: JavaScript

Nota

Estudiante	Escuela	Asignatura
Mariel Alisson Jara Mamani mjarama@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: I Código: 1702122

Laboratorio	Tema	Duración
03	JavaScript	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 30 Abril 2024	Al 04 Mayo 2024

## Índice

<b>1. Tarea</b>	<b>3</b>
1.1. Ejercicio de Prueba . . . . .	3
1.1.1. Commits . . . . .	3
1.1.2. Pruebas . . . . .	3
1.2. Ejercicio 1 . . . . .	3
1.2.1. Commits . . . . .	3
1.2.2. Código . . . . .	4
1.2.3. Pruebas . . . . .	4
1.3. Ejercicio 2 . . . . .	5
1.3.1. Commits . . . . .	5
1.3.2. Código . . . . .	5
1.3.3. Diseño Responsivo . . . . .	6
1.3.4. Pruebas . . . . .	6
1.4. Ejercicio 3 . . . . .	6
1.4.1. Commits . . . . .	7
1.4.2. Código . . . . .	7
1.4.3. Diseño Responsivo . . . . .	8
1.4.4. Pruebas . . . . .	9
<b>2. Investigación</b>	<b>10</b>
2.1. JS Ofuscador . . . . .	10
2.1.1. Importar la herramienta en este caso: Obfuscador.io . . . . .	10
2.1.2. Función whiteSpace del HangMan . . . . .	10
2.1.3. Función ofuscada . . . . .	10
2.1.4. URL del código ofuscado para el funcionamiento del juego. . . . .	10
<b>3. Entregables</b>	<b>11</b>
<b>4. URL de Repositorio de Git Hub</b>	<b>11</b>
<b>5. Estructura de laboratorio 03</b>	<b>11</b>
<b>6. Rúbrica</b>	<b>12</b>
<b>7. Referencias</b>	<b>12</b>

## 1. Tarea

Todos los ejercicios se realizaron exclusivamente con JavaScript puro, incluyendo la estructura HTML.

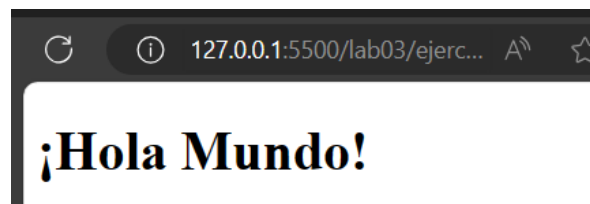
### 1.1. Ejercicio de Prueba

- Programar en JavaScript sobre una página web html básica (Se crea un contenedor y se pone un h1).

#### 1.1.1. Commits



#### 1.1.2. Pruebas




### 1.2. Ejercicio 1

- Cree un teclado random para banca por internet.


#### 1.2.1. Commits



**Ejercicio 02 completed**
56ee054

 Alsnj20 committed 2 hours ago

**Ejercicio 02: estructura generado con js**
8e31ac6

 Alsnj20 committed 4 hours ago

### 1.2.2. Código



- Creación de funciones para reutilizar código: createElement, createHeader, createBtn, generateStructureHTML, createFooter.
- **createElement():** Generalización de una función para reusar código, se encarga de crear los elementos y recibe tres parámetros: el tipo de elemento, un objeto con los atributos del elemento y el contenido del elemento.

```
const createElement = (tag, atributte = {}, content = '') => {
  const element = document.createElement(tag);
  for (let key in atributte) {
    element.setAttribute(key, atributte[key]);
  }
  element.innerHTML = content;
  return element;
}
```

- Forma de Uso

```
const h1 = createElement('h1', {}, ' Usted se encuentra en una <span>zona segura</span>');
```

### 1.2.3. Pruebas

 **Usted se encuentra en una zona segura**

Seleccione:

Número de tarjeta:

Tipo y N° Documento:

Ingresar tu clave usando el teclado virtual:









1	2	3
4	5	6
7	8	9
0	LIMPIAR	

### 1.3. Ejercicio 2

- 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.

#### 1.3.1. Commits

<b>metodo insertBefore en la ayuda de hacer la simulación de una pila</b>	e4b5251			
 Alsnj20 committed yesterday				
<b>Funcionalidad con eval ejercicio 02</b>	cf63532			
 Alsnj20 committed yesterday				
<b>btn equal del ejercicio 02 grid column</b>	f2ce30d			
 Alsnj20 committed yesterday				
<b>Estilo css del ejercicio 02</b>	6d90354			
 Alsnj20 committed yesterday				
<b>Actualizacion de la estructura ejercicio02</b>	0665c58			
 Alsnj20 committed yesterday				
<b>Asignando el primer estilo al ejercicio 02</b>	1a978c6			
 Alsnj20 committed yesterday				

<b>Ejercicio 02 completed</b>	56ee054			
 Alsnj20 committed 2 hours ago				
<b>Ejercicio 02: estructura generado con js</b>	8e31ac6			
 Alsnj20 committed 4 hours ago				

#### 1.3.2. Código

- Creacion de funciones para reutilizar código: `createElement`, `createBtns`, `createHistory`, `generateStructureHTML`, `createFooter`.
- **Inserción del contenido generado en el cuerpo del documento**

```
document.body.append(generateStructureHTML(), createFooter());}
```

- **Método 'insertBefore':** Este método fue de ayuda al momento de almacenar el historial, se utiliza para insertar un nuevo elemento antes.

```
pilaContent.insertBefore(divItem(operation), pilaContent.firstChild);
```

### 1.3.3. Diseño Responsivo

- **Propiedades para el Botón Igual:** Este estilo se aplica en específico: 'grid-column: 5/6; y grid-row: 4/6' se encargan de posicionar el botón en una cuadrícula CSS, ocupa la quinta columna y abarca las filas 4 y 5.

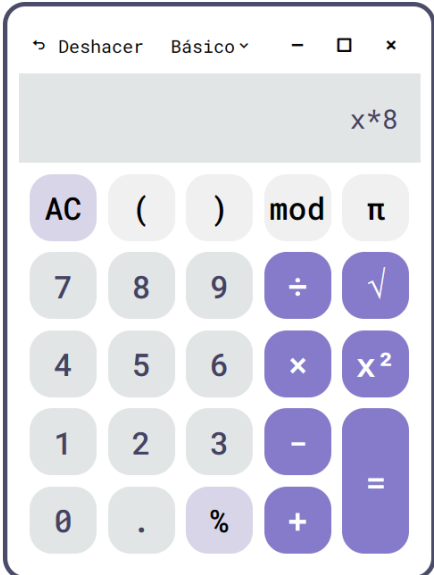
```
#equal {
  height: auto;
  grid-column: 5/6;
  grid-row: 4/6;
}
```

- **Media Query:** Esta media query nos permite manejar el diseño sin que haya colapsos de ancho o alto en relación al tamaño de la pantalla.

```
@media (max-width: 799px) {
  main {
    padding: 4%;
    grid-template-columns: repeat(1, 1fr);
    row-gap: 20px;
    height: auto;
  }
}
```

### 1.3.4. Pruebas

## Calculator



## Historial

16
x*8
2
x=2
81
(5+4)*9

### 1.4. Ejercicio 3

- Cree una versión de el juego 'el ahorcado' que grafique con canvas paso a paso desde el evento

onclick() de un botón.

#### 1.4.1. Commits

<b>Ejercicio 03 completed</b> Alsnoj20 committed 5 hours ago	616cfad			
<b>logica de hangMan completed</b> Alsnoj20 committed 5 hours ago	238cea8			
<b>Añadiendo estilo, usando una api para que las lineas se vean dibujadas</b> Alsnoj20 committed 10 hours ago	4d8eac3			
<b>evento onclick</b> Alsnoj20 committed 12 hours ago	b856d6e			
<b>logica del canvas para el hangman</b> Alsnoj20 committed 12 hours ago	58d9305			
<b>draw image canva</b> Alsnoj20 committed 14 hours ago	93316f0			
<b>Algunos comandos basicos para canvas</b> Alsnoj20 committed 14 hours ago	a60d10f			
Commits on May 17, 2024				
<b>draw a circle</b> Alsnoj20 committed yesterday	116017f			
<b>learn Canvas JS</b> Alsnoj20 committed yesterday	739eda6			

#### 1.4.2. Código

- **Estructura HTML** Creación de funciones para reutilizar código: createElement, createKeyboard, generateStructureHTML.
- **Funciones del Juego** Creación de funciones para el juego del ahorcado: startGame, resetGame, playGame, metodos auxiliares como: randomWord, whiteSpace, checkWord, restartAttemp, etc.
- **Uso de Canvas:**El 'canvas' es un elemento HTML que proporciona un lienzo rectangular o cuadrado en el cual podemos dibujar gráficos, imágenes y otros elementos mediante coordenadas. Para esta tarea, hemos utilizado la API rough.js, que nos permite crear líneas y otros elementos gráficos con un estilo de dibujo a mano alzada. A continuación se muestra el uso de Canvas.
- Esta línea selecciona el canvas del documento HTML utilizando su ID.

```
const itemCanvas = document.querySelector('#itemCanvas');
```

- Se obtiene el contexto de dibujo en 2D del canvas. Permitiendo hacer dibujos, líneas, etc.

```
const datactx = itemCanvas.getContext('2d');
```

- Se crea una instancia de canvas utilizando la biblioteca rough.js, que se inicializa con el canvas HTML seleccionado.

```
const roughCanvas = rough.canvas(itemCanvas);
```

- Lógica Implementada

```
switch (step) {  
  case 0:  
    // Estructura principal  
    roughCanvas.line(50, 450, 200, 450); // Base  
    roughCanvas.line(125, 450, 125, 50); // Poste vertical  
    roughCanvas.line(125, 50, 280, 50); // Poste horizontal  
    roughCanvas.line(280, 50, 280, 100); // Cuerda  
    roughCanvas.circle(280, 140, 80); // Cabeza  
    break;  
  ...  
  case 6:  
    roughCanvas.circle(280, 140, 80); // Cabeza  
    // Ojo derecho  
    roughCanvas.line(270, 130, 260, 120);  
    roughCanvas.line(260, 130, 270, 120);  
    // Ojo izquierdo  
    roughCanvas.line(290, 130, 300, 120);  
    roughCanvas.line(300, 130, 290, 120);  
    // Boca  
    roughCanvas.line(270, 150, 300, 145, 4);  
    // Lengua  
    roughCanvas.arc(280, 148, 20, 10, 0, 0.6 * Math.PI, true);  
    break;  
}
```

### 1.4.3. Diseño Responsivo

- **Media Query:** Esta media query se activa cuando el ancho de la pantalla es de 799 píxeles o menos de la siguiente forma: 'body' ajusta la altura automáticamente, '.container' cambia la cuadrícula a una sola columna, '.content' ajusta la altura al contenido máximo, 'word' fija la altura del elemento word.

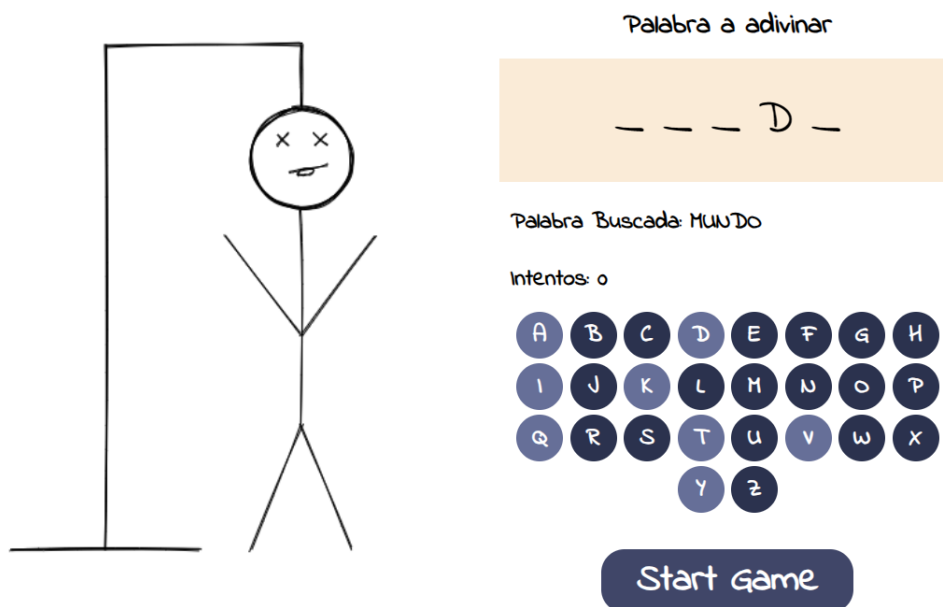
```
@media (max-width: 799px) {  
  body {  
    height: auto;  
  }  
  .container {  
    grid-template-columns: repeat(1, 1fr);  
  }  
}
```



```
.content {
  height: max-content;
}
#word {
  height: 100px;
}
}
```

#### 1.4.4. Pruebas

## HANG MAN



## 2. Investigación

### 2.1. JS Ofuscador

Un ofuscador de JavaScript es una herramienta que se utiliza para transformar el código fuente de JavaScript de manera que sea más difícil de entender para los humanos, pero que siga siendo funcional para los navegadores y motores de JavaScript. Esto se logra mediante técnicas como cambiar los nombres de las variables y funciones a versiones más cortas, eliminar comentarios y espacios en blanco, y reorganizar el código de manera que sea más difícil de seguir la lógica del programa.

#### 2.1.1. Importar la herramienta en este caso: Obfuscador.io

En este caso puede ser por npm o desde un link. Para el ejemplo se uso el link.

```
<script  
  src="https://cdn.jsdelivr.net/npm/javascript-obfuscator/dist/index.browser.js"></script>
```

#### 2.1.2. Función whiteSpace del HangMan

```
const whiteSpace = (word) => {  
  let whiteSpace = '';  
  for (let i = 0; i < word.length; i++) {  
    whiteSpace += ' ';  
  }  
  return whiteSpace;  
}
```

#### 2.1.3. Función ofuscada

```
function _0x2ddb(_0xaa7a23,_0x2e350a){const _0x81cc57=_0x81cc();return  
_0x2ddb=function(_0x2ddb0,_0x24bbb5){_0x2ddb0=_0x2ddb0-0x7d;let  
_0x11fd08=_0x81cc57[_0x2ddb0];return  
_0x11fd08;},_0x2ddb(_0xaa7a23,_0x2e350a);}function _0x81cc(){const  
_0x1a792b=['141106gowoef','573652aevNcU','712116ljvWcQ','2455180eZW0ir',  
'321282RfmHKH','7145703UDOpoS','248tWsPPi','539777sfkETW','length',  
'3jTSlyS'];_0x81cc=function(){return _0x1a792b;};return _0x81cc();}  
(function(_0x48ed05,_0x336d6d){const _0x5a263a=_0x2ddb,_0x3288fd=_0x48ed05();  
while (!![]){try{const _0x534b6b=parseInt(_0x5a263a(0x80))/0x1+  
parseInt(_0x5a263a(0x85))/0x2+parseInt(_0x5a263a(0x82))/0x3*  
(parseInt(_0x5a263a(0x84))/0x4)+-parseInt(_0x5a263a(0x86))/0x5+-  
parseInt(_0x5a263a(0x7d))/0x6+parseInt(_0x5a263a(0x83))/0x7*(-  
parseInt(_0x5a263a(0x7f))/0x8)+parseInt(_0x5a263a(0x7e))/0x9;  
if(_0x534b6b===_0x336d6d)break;else _0x3288fd['push'](_0x3288fd['shift']  
( ));}catch(_0x4cf7e0){_0x3288fd['push'](_0x3288fd['shift'] ( ));}}}  
(_0x81cc,0xa20b6));const whiteSpace=_0x5f1207=>{const _0x1d2037=_0x2ddb;let  
_0x18a769='';for(let _0x520ce5=0x0;_0x520ce5<_0x5f1207[_0x1d2037(0x81)];_0x520ce5++)  
{_0x18a769+=' '};return _0x18a769;};
```

#### 2.1.4. URL del código ofuscado para el funcionamiento del juego.

- <https://github.com/Alsnj20/pw2-24a/blob/main/lab03/js/ejercicio03.min.js>

### 3. Entregables

- Informe de laboratorio.
- Archivos correspondiente en el Repositorio.
- URL del Repositorio.

### 4. URL de Repositorio de Git Hub

- <https://github.com/Alsnj20/pw2-24a>

### 5. Estructura de laboratorio 03

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab03/
|---/css
|   |---ejercicio01.css
|   |---ejercicio02.css
|   |---ejercicio03.css
|---/exercises
|---/img
|   |---bancoNacion.png
|   |---candado.png
|   |---logoRedVirtual.jpg
|---/js
|   |---ejercicio00.js
|   |---ejercicio01.js
|   |---ejercicio02.js
|   |---ejercicio03.js
|   |---ejercicio03.min.js
|---/latex
|   |--- linopinto_pw2_24a_lab03.tex
|   |--- linopinto_pw2_24a_lab03.pdf
|---.gitignore
|---ejercicio00.html
|---ejercicio01.html
|---ejercicio02.html
|---ejercicio03.html
|--- README.md
```

## 6. Rúbrica

Tabla: 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 observación)	4	×	4	
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación)	4	×	4	
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	×	4	
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación)	2	×	2	
7. Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2	×	1	
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	
<b>Total</b>		20	Completo	18	

## 7. Referencias

- <https://github.com/>
- <https://git-scm.com/>
- [https://www.w3schools.com/graphics/canvas\\_intro.asp](https://www.w3schools.com/graphics/canvas_intro.asp)
- <https://roughjs.com/>
- <https://obfuscator.io/>