

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

## INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	PROGRAMACIÓN WEB 2				
TÍTULO DE LA PRÁCTICA:	JavaScript				
NÚMERO DE PRÁCTICA:	N° 3	AÑO LECTIVO:	Segundo	NRO. SEMESTRE:	III
FECHA DE PRESENTACIÓN	07/10/2024	HORA DE PRESENTACIÓN	20:12		
INTEGRANTE (s) DONNY MOISES MARA MAMANI ANTONY JUAN TAPIA HUAMANTUMA				NOTA (0-20)	Nota colocada por el docente
DOCENTE(s): LINO JOSE PINTO OPPE					

## RESULTADOS Y PRUEBAS

### I. EJERCICIOS RESUELTOS:

#### 1. Ejercicio 1. Cree un teclado random para banca por internet.

##### - Commits

platilla html basica para el ejercicio del teclado	0b5db15	<>
MMaraP committed 18 hours ago		
Creando el script para la generacion de datos en el html	5ddf9f2	<>
MMaraP committed 16 hours ago		
script.js metodo para generar numeros randoms en un teclado	22ed86a	<>
MMaraP committed 16 hours ago		
generando el teclado	71098d6	<>
MMaraP committed 16 hours ago		
diseñando la pagina web	4b282cf	<>
MMaraP committed 15 hours ago		

##### - Código

```
document.addEventListener("DOMContentLoaded", () => {
  const headerDiv = document.createElement('div');
  headerDiv.className = 'header';

  const logoMultiRed = document.createElement('img');
  logoMultiRed.src = 'img/multiRed_Logo.jpg';
  logoMultiRed.alt = 'Multi Red Virtual';
  logoMultiRed.className = 'multired_logo';

  const logoBanco = document.createElement('img');
  logoBanco.src = 'img/logo-banco_de_la_nacion.png';
  logoBanco.alt = 'Banco De La Nacion';
  logoBanco.className = 'banco_logo';

  headerDiv.appendChild(logoMultiRed);
  headerDiv.appendChild(logoBanco);
});
```

Se crea un elemento <div> con la clase header.

Se crean dos elementos <img> para los logotipos de "Multi Red Virtual" y "Banco De La Nacion". Los logotipos se configuran con sus respectivas rutas de imagen y texto alternativo.

Los logotipos se agregan como hijos del elemento <div> del

encabezado. Creación del encabezado (headerDiv):

```
const paragraph = document.createElement('p');
paragraph.innerHTML = "<i class='bx bxs-lock'></i>USTED SE ENCUENTRA EN UNA ZONA SEGURA";

const containerDiv = document.createElement('div');
containerDiv.className = 'container';

const formContainerDiv = document.createElement('div');
formContainerDiv.className = 'container-formulario';

const form = document.createElement('form');
form.id = 'formulario';
form.action = '';
```

Se crea un elemento <p> con un mensaje de texto dentro que indica que el usuario se encuentra en una zona segura.

El mensaje está marcado como contenido HTML, lo que permite la inserción de un ícono de cerradura. Creación de contenedores (containerDiv y formContainerDiv):

Se crean dos elemntos <div> con las clases container y container-formulario respectivamente. Creación del formulario (form):

Se crea un formulario con el id 'formulario' y sin atributo action. Creación de elementos del formulario:

```
const seleccionDiv = document.createElement('div');
seleccionDiv.className = 'seleccion-banco';

const labelSeleccion = document.createElement('label');
labelSeleccion.htmlFor = 'seleccion';
labelSeleccion.textContent = 'Seleccione: ';

const selectSeleccion = document.createElement('select');
selectSeleccion.id = 'seleccion';

const optionSeleccion = document.createElement('option');
optionSeleccion.value = 'multired-global-debito';
optionSeleccion.textContent = 'Multired Global Débito';

selectSeleccion.appendChild(optionSeleccion);
seleccionDiv.appendChild(labelSeleccion);
seleccionDiv.appendChild(selectSeleccion);
```

	<p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<p style="text-align: center;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 4</p>

Se crean elementos de formulario como <select>, <input>, <label>, etc., para recopilar información del usuario como el tipo de tarjeta, número de tarjeta, tipo de documento, número de documento, clave de acceso, etc.

Configuración del teclado virtual (tecladoVirtualDiv):

```
const tecladoDiv = document.createElement('div');
tecladoDiv.className = 'teclado-banca';

const labelTeclado = document.createElement('label');
labelTeclado.htmlFor = 'clave';
labelTeclado.textContent = 'Ingrese su clave en el Teclado Virtual:';

const tecladoVirtualDiv = document.createElement('div');
tecladoVirtualDiv.id = 'teclado';

tecladoDiv.appendChild(labelTeclado);
tecladoDiv.appendChild(tecladoVirtualDiv);
```

Se crea un elemento <div> que actúa como el teclado virtual. Configuración del botón de envío (submitButton):

```
const submitButton = document.createElement('button');
submitButton.type = 'submit';
submitButton.textContent = 'INGRESAR';

form.appendChild(seleccionDiv);
form.appendChild(tarjetaDiv);
form.appendChild(tipoDocumentoDiv);
form.appendChild(tecladoDiv);
form.appendChild(claveAccesoDiv);
form.appendChild(submitButton);

formContainerDiv.appendChild(form);

containerDiv.appendChild(formContainerDiv);

document.body.appendChild(headerDiv);
document.body.appendChild(paragraph);
document.body.appendChild(containerDiv);
```

Se crea un botón de tipo 'submit' con el texto "INGRESAR". Agregación de elementos al DOM:

Finalmente, todos los elementos creados se agregan al DOM, primero el encabezado, luego el párrafo, y finalmente el contenedor del formulario.

```
document.addEventListener('DOMContentLoaded', (event) => {
  const formulario = document.getElementById('formulario');
  const teclado = document.getElementById('teclado');
  const clave = document.getElementById('clave');
```

Esta parte del código escucha el evento 'DOMContentLoaded', que se dispara cuando el DOM ha sido completamente cargado. Luego, obtiene referencias a tres elementos del DOM: el formulario, el contenedor del teclado virtual y el campo de entrada de la clave de acceso.

```
function generarNumerosRandom() {
  const numbers = Array.from({ length: 10 }, (_, i) => i);
  for (let i = numbers.length - 1; i > 0; i--) {
    const j = Math.floor(Math.random() * (i + 1));
    [numbers[i], numbers[j]] = [numbers[j], numbers[i]];
  }
  return numbers;
}
```

Esta función genera una matriz de números aleatorios del 0 al 9 utilizando el algoritmo de Fisher-Yates para mezclar aleatoriamente los números en la matriz.

```
function crearTeclado() {
  const numeros = generarNumerosRandom();
  teclado.innerHTML = '';

  numeros.forEach(number => {
    const button = document.createElement('button');
    button.type = 'button';
    button.textContent = number;
    button.addEventListener('click', (event) => {
      event.preventDefault();
      if (clave.value.length < 6) {
        clave.value += number;
      }
    });
    teclado.appendChild(button);
  });

  const clearButton = document.createElement('button');
```

Esta función crea dinámicamente los botones del teclado virtual y los agrega al contenedor del teclado. Además, agrega un evento de clic a cada botón para que al hacer clic, se agregue el número correspondiente al campo de entrada de la clave. También agrega un botón 'Limpiar' que borra el contenido del campo de la clave.

#### - PRUEBA



- 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.

#### - COMMITS

html basico para la calculadora	179d2a7	<>
MMaraP committed 1 hour ago		
script para generar la estructura del html	1f3e33c	<>
MMaraP committed 1 hour ago		
almacenamiento de las operaciones realizadas	1f3a598	<>
MMaraP committed 30 minutes ago		
agregar un valor a la operacion, actualizando el campo de entrada de ...	5d78aaf	<>
MMaraP committed 29 minutes ago		
limpiar el campo de entrada de la calculadora	c7a5fd3	<>
MMaraP committed 28 minutes ago		
calculando el resultando con eval()	a8b8f70	<>
MMaraP committed 24 minutes ago		

#### - CODIGO

	<p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<p style="text-align: center;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 7</p>

```
document.addEventListener("DOMContentLoaded", () => {

    const programaDiv = document.createElement('div');
    programaDiv.id = 'programa';

    const form = document.createElement('form');
    form.name = 'calculador';

    const inputDisplay = document.createElement('input');
    inputDisplay.type = 'text';
    inputDisplay.name = 'respuesta';
    inputDisplay.value = '';
    inputDisplay.autocomplete = 'off';
    inputDisplay.readOnly = true;
    form.appendChild(inputDisplay);
```

Este fragmento de código establece una escucha para el evento 'DOMContentLoaded', el cual se dispara cuando el DOM ha sido completamente cargado. Se crean elementos del DOM como un contenedor principal (programaDiv), un formulario (form) y un campo de entrada (inputDisplay) para mostrar la respuesta de la calculadora.

```
for (let i = 0; i < valores.length; i += 4) {
    const filaDiv = document.createElement('div');
    filaDiv.className = 'fila';

    for (let j = 0; j < 4; j++) {
        const boton = valores[i + j];
        const inputButton = document.createElement('input');
        inputButton.type = boton.type || 'button';
        inputButton.value = boton.value;
        inputButton.className = boton.class;

        if (boton.value === 'C') {
            inputButton.onclick = function() { limpiar(); };
        } else if (boton.value === '=') {
            inputButton.onclick = function() { resultado(); };
        } else {
```

Este bloque de código crea dinámicamente los botones de la calculadora y los agrega al formulario. Se itera sobre un conjunto predefinido de valores de botones, creando filas de botones.

Se asignan eventos a los botones según su valor. Para el botón 'C' se asigna la función limpiar(), para el botón '=' se asigna resultado(), y para los demás botones se asigna agregarValor() con el valor correspondiente.



```
let operacionActual = '';
let pila = [];

function agregarValor(value) {
    operacionActual += value;
    document.calculador.respuesta.value = operacionActual;
}

function limpiar() {
    operacionActual = '';
    document.calculador.respuesta.value = '';
}
```

operacionActual: Almacena la operación actual que se está realizando en la calculadora. pila: Almacena el historial de operaciones realizadas.

agregarValor(): Esta función agrega un valor al final de la operación actual y actualiza el campo de entrada de la calculadora con la operación actualizada.

Limpiar(): Esta función restablece la operación actual y limpia el campo de entrada de la calculadora.

```
function resultado() {
    try {
        let resultado = eval(operacionActual);
        pila.push(operacionActual + ' = ' + resultado);
        actualizarPila();
        operacionActual = '';
        document.calculador.respuesta.value = resultado;
    } catch (e) {
        document.calculador.respuesta.value = 'Error';
    }
}

function actualizarPila() {
    let listaPila = document.getElementById('lista-pila');
    listaPila.innerHTML = '';
    pila.forEach(function (operacion) {
        let li = document.createElement('li');
        li.textContent = operacion;
        listaPila.appendChild(li);
    });
}
```

Resultado(): Esta función calcula el resultado de la operación actual utilizando la función eval(). Agrega la operación actual y su resultado a la pila de operaciones.



	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>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: 9</p>

Llama a la función `actualizarPila()` para reflejar los cambios en la lista de operaciones. Limpia la operación actual y muestra el resultado en el campo de entrada de la calculadora.  
`actualizarPila()`: Esta función actualiza la lista de operaciones en la interfaz de usuario.

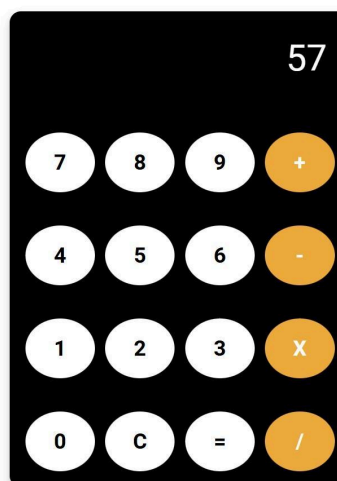
Borra el contenido anterior de la lista.

Itera sobre todas las operaciones en la pila y agrega cada una como un elemento de lista (`<li>`) a la lista de operaciones en el DOM.

#### - PRUEBA

Pila de operaciones:

$9+9 = 18$   
 $3*6 = 18$   
 $3/5 = 0.6$   
 $9+6*8 = 57$



- Ejercicio 3.** Cree una versión del juego “El ahorcado” que grafique con canvas paso a paso delde el evento onclick de un botón.

## - COMMITS

plantilla html para el ejercicio del ahorcado	96063e4	<>
script que genera el html	6cbf0ab	<>
array de palabras, seleccion aleatoria, inicializacion del contado...	d03a5b9	<>
funcion para ingresar una palabra al presionar enter	8d3a1d6	<>
Creando funcio para adivinar la palabra, verificando si el juego termino	ad93e4d	<>
obteniendo la palabra ingresada y convirtiela en mayusculas	f7ed7f1	<>

## - CÓDIGO

```
document.addEventListener("DOMContentLoaded", () => {

    const titulo = document.createElement('h1');
    titulo.textContent = 'Juego del Ahorcado';
    document.body.appendChild(titulo);


    const canvas = document.createElement('canvas');
    canvas.id = 'cuadro';
    canvas.width = 400;
    canvas.height = 400;
    document.body.appendChild(canvas);
});
```

Titulo: Se crea un elemento h1 que contiene el título del juego. El título se establece con el texto "Juego del Ahorcado".

El elemento h1 se agrega como hijo del cuerpo del documento.

Canvas: Se crea un elemento canvas que servirá como área de dibujo para el juego. Se establecen propiedades como el ID y las dimensiones del canvas.

El canvas se agrega como hijo del cuerpo del documento.

	<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;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 11</p>

```
const divInput = document.createElement('div');

const label = document.createElement('label');
label.htmlFor = 'letra';
label.textContent = 'Introduce una letra: ';
divInput.appendChild(label);

const input = document.createElement('input');
input.type = 'text';
input.id = 'letra';
input.maxLength = 1;
input.onkeypress = function(event) { teclaPresionada(event); };
divInput.appendChild(input);

document.body.appendChild(divInput);
```

Se crea un contenedor div para el campo de entrada y su etiqueta asociada. Se crea una etiqueta label que indica al usuario qué debe ingresar.

Se crea un campo de entrada input de tipo texto para que el usuario ingrese una letra. Se limita la longitud máxima del campo de entrada a 1 carácter.



Se agrega un evento onkeypress que llama a la función teclaPresionada() cuando se presiona una tecla en el campo de entrada.

El contenedor div con la etiqueta y el campo de entrada se agrega como hijo del cuerpo del documento.

```
const palabraParrafo = document.createElement('p');
palabraParrafo.id = 'palabra';
document.body.appendChild(palabraParrafo);

const mensajeParrafo = document.createElement('p');
mensajeParrafo.id = 'mensaje';
document.body.appendChild(mensajeParrafo);
```

Se crean dos elementos p para mostrar la palabra oculta y los mensajes del juego. Se establece un ID único para cada elemento.

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 12</p>

Ambos elementos se agregan como hijos del cuerpo del documento.

```
const script = document.createElement('script');
script.src = 'js/script.js';
document.body.appendChild(script);
```

Se crea un elemento script que carga el archivo JavaScript que contiene la lógica del juego.

Se establece la ruta del archivo JavaScript en la propiedad src.

El script se agrega como hijo del cuerpo del documento para que se ejecute y maneje la funcionalidad del juego.

```
const palabras = ["JAVASCRIPT", "PROGRAMACION", "CANVAS", "AHORCADO"];
let palabra = palabras[Math.floor(Math.random() * palabras.length)];
let palabraAdivinada = "_".repeat(palabra.length);
let intentos = 0;
const maxIntentos = 10;
let juegoTerminado = false;
```

Se define un array de palabras.

Se selecciona aleatoriamente una palabra del array.

Se inicializa una variable para representar la palabra a adivinar, inicialmente mostrando guiones bajos.

Se inicializa un contador de intentos y se establece el máximo de intentos permitidos. Se establece la palabra a adivinar en el elemento con el ID 'palabra' en el DOM.

```
function teclaPresionada(evento) {
  if (evento.key === 'Enter') {
    adivinarLetra();
  }
}
```

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 13

Esta función maneja el evento de presionar una tecla, permitiendo adivinar la letra cuando se presiona la tecla "Enter".

```
function adivinarLetra() {
    if (juegoTerminado) {
        return;
    }

    const inputLetra = document.getElementById('letra');
    const letra = inputLetra.value.toUpperCase();
    inputLetra.value = '';
    if (!letra || !/^[A-ZÑ$]/.test(letra)) {
        alert("Introduce una letra válida");
        return;
    }

    if (palabra.includes(letra)) {
        let nuevaPalabraAdivinada = '';
        for (let i = 0; i < palabra.length; i++) {
            if (palabra[i] === letra) {
                nuevaPalabraAdivinada += letra;
            } else {
                nuevaPalabraAdivinada += palabraAdivinada[i];
            }
        }
        palabraAdivinada = nuevaPalabraAdivinada;
        document.getElementById('palabra').textContent = palabraAdivinada;
    } else {
        intentos++;
        dibujarAhorcado(intentos);
    }
}
```

Esta función se encarga de adivinar la letra ingresada por el usuario. Verifica si el juego ha terminado antes de realizar cualquier acción. Obtiene la letra ingresada por el usuario y la convierte a mayúsculas. Valida si la letra ingresada es válida (una letra del alfabeto).

Compruebe si la letra esta presente en la palabra a adivinar

Si la letra está presente, actualiza la palabra adivinada con la letra correcta.

Si la letra no está presente, incrementa el contador de intentos y dibuja una parte del ahorcado.

Finalmente, comprueba si el juego ha terminado (ya sea ganando o perdiendo).

```
function dibujarAhorcado(intentos) {  
    const cuadro = document.getElementById('cuadro');  
    const dibujar = cuadro.getContext('2d');  
  
    if (intentos === 1) {  
        dibujar.beginPath();  
        dibujar.moveTo(50, 350);  
        dibujar.lineTo(150, 350);  
        dibujar.stroke();  
    } else if (intentos === 2) {  
        dibujar.beginPath();  
        dibujar.moveTo(100, 350);  
        dibujar.lineTo(100, 50);  
        dibujar.stroke();  
    } else if (intentos === 3) {
```

Esta función se encarga de dibujar diferentes partes del ahorcado en el canvas según el número de intentos realizados.

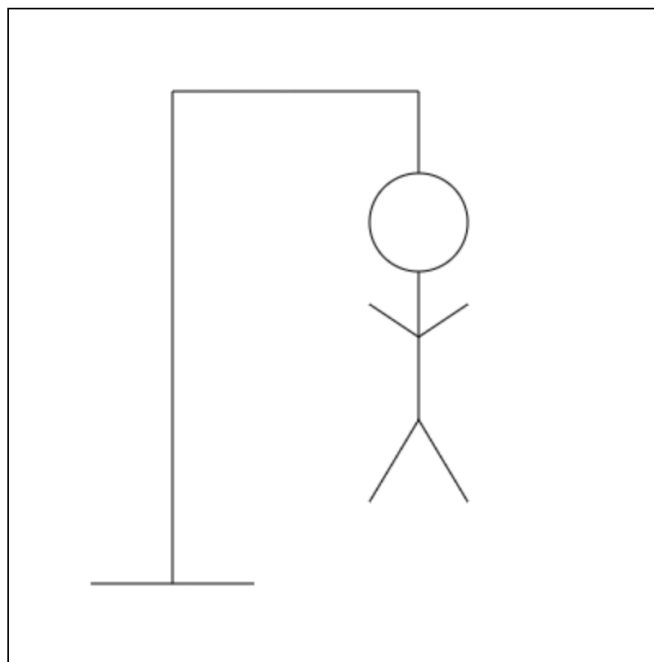
Utiliza el contexto 2D del canvas para realizar los dibujos.

Dibuja partes del ahorcado progresivamente según el número de intentos realizados.

- **PRUEBA**

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>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: 15</p>

## Juego del Ahorcado



Introduce una letra:

CAN\_AS

TE AHORCASTE!!!. La palabra era: CANVAS

## II. CUESTIONARIO:

*Explique una herramienta para ofuzcar código JavaScript.*

*JavaScript Obfuscator: JavaScript Obfuscator es una herramienta que oculta y protege el código JavaScript al convertirlo en una forma ilegible para los humanos, mientras sigue siendo funcional para los navegadores web. Utiliza técnicas avanzadas para cambiar nombres de variables y funciones, elimina comentarios y espacios en blanco, y protege contra la ingeniería inversa. Se integra fácilmente en los flujos de trabajo de desarrollo y ayuda a mejorar la seguridad y la integridad del código en aplicaciones web. Un obfuscador de JavaScript es una herramienta que transforma el código fuente de JavaScript en una forma ofuscada o encriptada. El objetivo principal de un obfuscador es dificultar la comprensión y el análisis del código por parte de terceros, sin afectar su funcionalidad. La ofuscación del código implica cambiar los nombres de variables y funciones, reemplazar cadenas legibles por cadenas codificadas y alterar el flujo de control del programa.*



Copy &amp; Paste JavaScript Code

Upload JavaScript File

Output

```
1 const palabras = ["JAVASCRIPT", "PROGRAMACION", "CANVAS", "AHORCADO"];
2 let palabra = palabras[Math.floor(Math.random() * palabras.length)];
3 let palabraAdivinada = "_".repeat(palabra.length);
4 let intentos = 0;
5 const maxIntentos = 10;
6 let juegoTerminado = false;
7
8 document.getElementById('palabra').textContent = palabraAdivinada;
```

Obfuscate

Copy &amp; Paste JavaScript Code

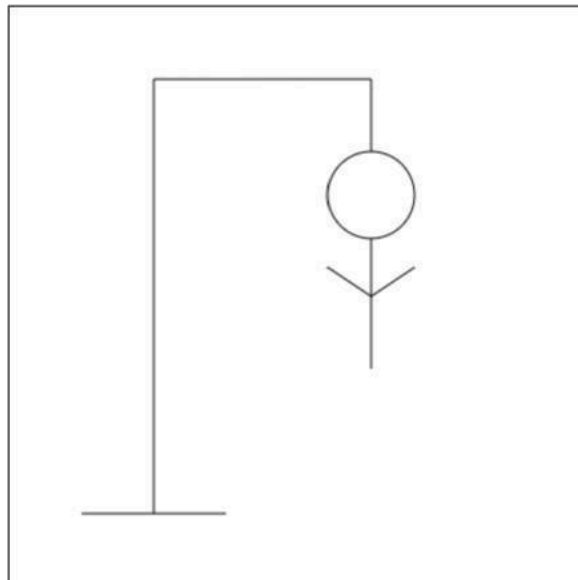
Upload JavaScript File

Output

```
const _0x5bd5d5=_0x2211;function _0x2211(_0x48c137,_0x1d0177){const _0x1ebcbb=_0x1ebc();return _0x2211=function(_0x2211c3,_0x4568b8){_0x2211c3=_0x2211c3-0x190;let _0x35ea79=_0x1ebcbb[_0x2211c3];return _0x35ea79;}_0x2211(_0x48c137,_0x1d0177);}(function(_0x2174aa,_0x44c7a0){const _0x20f0d8=_0x2211,_0x9360cb=_0x2174aa();while(![]){try{const _0x99cad7=-parseInt(_0x20f0d8(0x193))/0x1+-parseInt(_0x20f0d8(0x1a4))/0x2*(parseInt(_0x20f0d8(0x197))/0x3)+-parseInt(_0x20f0d8(0x1a2))/0x4*(parseInt(_0x20f0d8(0x1a9))/0x5)+parseInt(_0x20f0d8(0x1a5))/0x6+-parseInt(_0x20f0d8(0x196))/0x7+parseInt(_0x20f0d8(0x19c))/0x8+parseInt(_0x20f0d8(0x1b2))/0x9;if(_0x99cad7===_0x44c7a0)break;else _0x9360cb['push'](_0x9360cb['shift']());}catch(_0x778d0a){_0x9360cb['push'](_0x9360cb['shift']());}})(_0x1ebc,0x55cef));const palabras=['JAVASCRIPT',_0x5bd5d5(0x198),_0x5bd5d5(0x1a1),_0x5bd5d5(0x1af)];let palabra=palabras[Math[_0x5bd5d5(0x1a7)]](Math[_0x5bd5d5(0x194)])(Math[_0x5bd5d5(0x1a9)]);palabraAdivinada=".".repeat(palabra.length);intentos=0;const maxIntentos=10;let juegoTerminado=false;document.getElementById('palabra').textContent=palabraAdivinada;
```

```
const script = document.createElement('script');
script.src = 'js/script_obfuscated.js';
document.body.appendChild(script);
```

## Juego del Ahorcado



Introduce una letra:

JAVASCRIPT

¡Felicidades! ¡Ganaste!

*Estrcutra de escritorio de archivos del laboratorio (Nivel 4)*

```
C:.\n  estructura_escritorio.txt\n  README.md\n+---LABORATORIO_01\n  |   Calculator.class\n  |   Calculator.java\n  |   INFORME - LABORATORIO_01.pdf\n+---LABORATORIO_03\n  |   estructura_escritorio.txt\n  |   +---Ahorcado\n  |   |   index.html\n  |   |   +---css\n  |   |   |   styles.css\n  |   |   +---js\n  |   |   |   html.js\n  |   |   |   script.js\n  |   |   |   script_obfuscated.js\n  |   +---Calculadora\n  |   |   index.html\n  |   |   +---css\n  |   |   |   styles.css\n  |   |   +---js\n  |   |   |   html.js\n  |   |   |   script.js\n  |   +---Teclado_Random\n  |   |   index.html\n  |   |   +---css\n  |   |   |   styles.css\n  |   |   +---img\n  |   |   |   logo-banco_de_la_nacion.png\n  |   |   |   multiRed_Logo.jpg\n  |   +---script\n  |   |   html.js\n  |   |   script.js
```

**REFERENCIAS Y BIBLIOGRAFÍA**<https://github.com/LINOPINTO2023/PWII-2024B/tree/TechSquad><https://github.com/LINOPINTO2023/PWII-2024B.git><https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API>

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 18

### RUBRICA PARA EL CONTENIDO DEL INFORME Y DEMOSTRACIÓN

Contenido y demostración		Puntos	Checklis t	Estudiant e	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	<b>X</b>	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	<b>X</b>	3	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	<b>X</b>	1	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	<b>X</b>	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	<b>X</b>	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	<b>X</b>	1	
7. Ortografía	El documento no muestra errores ortográficos.	2	<b>X</b>	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	<b>X</b>	3	
<b>TOTAL</b>		<b>20</b>		<b>15</b>	