
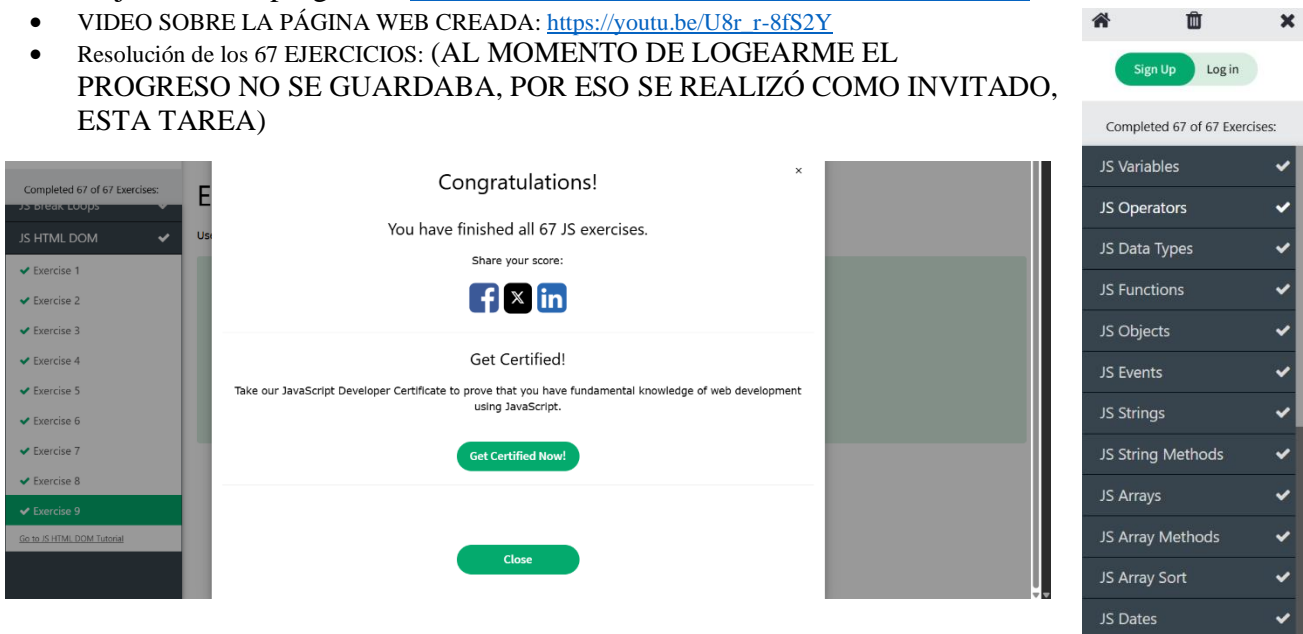
	UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 1

INFORME DE LABORATORIO

N° 03

INFORMACIÓN BÁSICA					
ASIGNATURA:	<i>Programación Web 2.</i>				
TÍTULO DE LA PRÁCTICA:	<i>JavaScript.</i>				
NÚMERO DE PRÁCTICA:	<i>03</i>	AÑO LECTIVO:	<i>2025-A</i>	NRO. SEMESTRE:	<i>III</i>
FECHA DE PRESENTACIÓN	<i>27/04/2025</i>	HORA DE PRESENTACIÓN	<i>18:30:00</i>		
INTEGRANTE (s) <i>Subia Huaicane Edson Fabricio</i>				NOTA (0-20)	<i>Nota colocada por el docente</i>
DOCENTE(s): <i>Ing. Carlo Corrales</i>					

RESULTADOS Y PRUEBAS	
<ul style="list-style-type: none"> EJERCICIOS RESUELTOS: Cree un Proyecto ... En este enlace se encuentra en el repositorio y los commits que se realizaron para la creación y/o mejora de este programa: https://github.com/Q3son/Pweb2---Edson-Subia.git VIDEO SOBRE LA PÁGINA WEB CREADA: https://youtu.be/U8r_r-8fS2Y Resolución de los 67 EJERCICIOS: (AL MOMENTO DE LOGEARME EL PROGRESO NO SE GUARDABA, POR ESO SE REALIZÓ COMO INVITADO, ESTA TAREA) 	

1. Descripción General del Sistema

El Laboratorio 03 implementa un portal interactivo con 7 ejercicios de JavaScript presentados mediante un innovador sistema orbital. El proyecto combina:

- 1.1. **Frontend moderno:** HTML5 semántico, CSS3 con animaciones y JavaScript ES6+
- 1.2. **Arquitectura modular:** Separación clara de responsabilidades (HTML, CSS, JS)
- 1.3. **Diseño responsive:** Adaptable a móviles, tablets y desktop
- 1.4. **Interacciones avanzadas:** Efectos hover, rotaciones 3D y transiciones suaves

2. Componentes Clave del Sistema

2.1. Sistema Orbital (Innovación Principal)

```
// orbital.js - Núcleo del sistema
function animate(currentTime) {
    const deltaTime = currentTime - lastTime;
    angle += rotationSpeed * (deltaTime / 16); // Animación a 60fps
    orbit.style.transform = `translate(-50%, -50%) rotate(${angle}deg)`;
    requestAnimationFrame(animate);
}
```

2.1.1. Funcionamiento:

- Cálculo preciso de posiciones angulares
- Animación fluida con requestAnimationFrame
- Control de velocidad basado en tiempo real (deltaTime)

2.2. Cards Interactivas

```
/* orbital.css - Efectos visuales */
.satellite:hover {
    transform: translate(-50%, -50%) scale(1.15) !important;
    box-shadow: 0 0 30px var(--satellite-color);
    z-index: 20;
}
```

2.2.1. Características

- Efectos hover con transformaciones 3D.
- Tooltips dinámicos con información.
- Sistema de capas (z-index) para superposiciones.

2.3. Resolución de Ejercicios (EN EL REPOSITORIO Y EL VÍDEO SE NOTAN MUCHO MEJOR EL CÓDIGO EN GENERAL Y EL BUEN FUNCIONAMIENTO DE NUESTRA PÁGINA WEB.

Ejercicio	Técnica Implementada	Dificultad	Solución Clave
Invertir Texto	<code>string.split().reverse().join()</code>	Baja	Validación de input
Día de la Semana	Array de días + <code>Date.getDay()</code>	Baja	Internacionalización
Días Arequipa	Cálculo entre fechas	Media	Manejo de timezone

Código Meet	Regex y manipulación de strings	Media	Validación de URL
Suma Tabla	Generación dinámica de DOM	Alta	Delegación de eventos
Editor Texto	Modificación CSS via JS	Media	Controles de rango
Calculadora	Switch con operadores	Alta	Validación de operandos

2.3.1. Invertir Texto

```
function invertirTexto() {
    const texto = document.getElementById('textoInput').value;
    const textoInvertido = texto.split('').reverse().join('');
    document.getElementById('resultado').textContent = textoInvertido;
}
```

- Funcionamiento
 - ✓ split("") convierte el string en array
 - ✓ reverse() invierte el orden del array
 - ✓ join("") convierte el array en string
 - ✓ **Validación:** Acepta cualquier caracter Unicode

2.3.2. Día de la Semana

```
function obtenerDiaSemana(fecha = new Date()) {
    const dias = ['Domingo', 'Lunes', 'Martes', 'Miércoles',
        'Jueves', 'Viernes', 'Sábado'];
    return dias[fecha.getDay()];
}
```



- Funcionamiento
 - ✓ getDay() devuelve índice 0-6 (0=Domingo)
 - ✓ Array con nombres localizados
 - ✓ **Mejora:** Acepta fecha personalizada o usa fecha actual

2.3.3. Días para las Fiestas de Arequipa

```
function diasParaArequipa() {
    const hoy = new Date();
    const año = hoy.getFullYear();
    let diaArequipa = new Date(año, 7, 15); // 15 de agosto

    if (hoy > diaArequipa) {
        diaArequipa = new Date(año + 1, 7, 15);
    }

    const diff = diaArequipa - hoy;
    return Math.ceil(diff / (1000 * 60 * 60 * 24));
}
```

	<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;">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: 4</p>

- Funcionamiento
 - ✓ **Cálculo preciso:**
 $1000 \text{ ms} \times 60 \text{ s} \times 60 \text{ min} \times 24 \text{ h} = 1 \text{ día}$
 - ✓ **TimeZone:** Usa la zona horaria local del navegador y la actualiza según la región asignada.

2.3.4.Código de Google Meet

```
function extraerCodigoMeet(url) {
  const partes = url.split('/');
  let codigo = partes[partes.length - 1];
  codigo = codigo.split('?')[0]; // Elimina parámetros
  return codigo.replace(/-/g, ''); // Elimina guiones
}
```



- Funcionamiento
 - ✓ Maneja la función Split y reemplaza los guiones por espacios, para así devolver el código de la URL ingresada.
 - ✓ **Ejemplo:**
 Input: meet.google.com/abc-def-ghi?auth=0
 Output: abcdefghi
 - ✓ **Robustez:** Maneja URLs con o sin parámetros.

2.3.5.Suma Tabla Dinámica

```
function generarTabla(filas) {
  const tabla = [];
  for (let i = 0; i < filas; i++) {
    tabla.push(Math.floor(Math.random() * 100) + 1);
  }
  return tabla;
}

function calcularSuma(tabla) {
  return tabla.reduce((sum, num) => sum + num, 0);
}
```

- Funcionamiento
 - ✓ Generación de números aleatorios 1-100.
 - ✓ Uso de reduce para suma eficiente.
 - ✓ **DOM Dinámico:** Crea tabla HTML basada en array.

	<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;">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: 5</p>

2.3.6. Página1.html (Editor de texto)

```
let tamañoActual = 16;

function cambiarTamaño(accion) {
  const texto = document.getElementById('textoEditable');
  tamañoActual = accion === 'aumentar' ?
    Math.min(tamañoActual + 2, 32) :
    Math.max(tamañoActual - 2, 10);
  texto.style.fontSize = `${tamañoActual}px`;
}

function cambiarColor(color) {
  document.getElementById('textoEditable').style.color = color;
}
```



- Funcionamiento
 - ✓ **Controles:**
 - ❖ Tamaño entre 10px y 32px.
 - ❖ Colores predefinidos (rojo, azul, verde, negro).
 - ✓ **Transiciones CSS:** Efectos suaves al cambiar propiedades.

2.3.7. Página2.html (Calculadora multifuncional)

```
function calcular(operador, a, b) {
  a = Number(a);
  b = Number(b);

  switch (operador) {
    case '+': return a + b;
    case '-': return a - b;
    case '*': return a * b;
    case '/': return b !== 0 ? a / b : 'Error';
    case '%': return a % b;
    case '**': return a ** b;
    case '&': return a & b;
    case '|': return a | b;
    case '^': return a ^ b;
    case '<<': return a << b;
    case '>>': return a >> b;
    case '&&': return Boolean(a && b);
    case '||': return Boolean(a || b);
    default: return 'Operador inválido';
  }
}
```

- Características
 - ✓ Soporta operaciones aritméticas, lógicas y de bits.
 - ✓ Validación de división por cero.
 - ✓ Conversión automática a números.

	<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;">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: 6</p>

- Cada ejercicio sigue el principio de responsabilidad única y está diseñado para:
 - ✓ Ser autónomo (no depende de variables globales).
 - ✓ Tener una interfaz clara.
 - ✓ Manejar errores adecuadamente.
 - ✓ Ser eficiente en términos de rendimiento.

En la siguiente sección mostraré el index.html código fuente y ejecución de la versión actual del código fuente del programa, trabajado en Visual Studio. (El código fuente versionado por commits y su ejecución se visualizan mucho mejor en el repositorio y el video al inicio de este informe).

```



index.html X
Lab_03 > index.html > html > body > div.exercises-grid > a.exercise-card
You: 8 hours ago | 1 author (You)
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Laboratorio PWeb2 | Edson Subia</title>
7   <link rel="stylesheet" href="assets/css/main.css">
8   <link rel="stylesheet" href="assets/css/orbital.css">
9   <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;600;700&display=swap" rel="stylesheet">
10 </head>
11 <body>
12   <div class="universe-bg"></div>
13
14   <header class="main-header">
15     <div class="header-content">
16       
17       <h1>Laboratorio PWeb2</h1>
18       
19     </div>
20     <p class="subtitle">Ejercicios de Programación Web 2 - Laboratorio 03</p>
21   </header>
22   <div class="orbital-system" id="orbitalSystem"></div>
23
24

```

```

25   <div class="exercises-grid">
26     <a href="ejercicios/texto_invertido.html" class="exercise-card">
27       
28       <h3>Invertir Texto</h3>
29     </a>
30
31     <a href="ejercicios/día_Semana.html" class="exercise-card">
32       
33       <h3>Día de la Semana</h3>
34     </a>
35
36     <a href="ejercicios/días_Arequipa.html" class="exercise-card">
37       
38       <h3>Día de Arequipa</h3>
39     </a>
40     <a href="ejercicios/código_de_meet.html" class="exercise-card">
41       
42       <h3>Código de Google Meet</h3>
43     </a>
44     <a href="ejercicios/suma_tabla.html" class="exercise-card">
45       
46       <h3>Suma en tabla dinámica</h3>
47     </a>
48     <a href="ejercicios/Página1.html" class="exercise-card">
49       
50       <h3>Cambiar propiedades de texto</h3>
51     </a>
52     <a href="ejercicios/Página2.html" class="exercise-card">
53       
54       <h3>Calculadora</h3>
55     </a>
56   </div>
57
58   <footer class="main-footer">
59     <p>© <span id="current-year">2025</span> - Edson Subia - UNSA - Ingeniería de Sistemas</p>
60   </footer>
61
62   <script src="assets/js/main.js"></script>
63   <script src="assets/js/orbital.js"></script>
64 </body>
65 </html>

```

	<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: 7</p>

Cuestionario

1. ¿Cómo se resolvió el problema de desaparición de elementos durante el hover?

Implementamos:

- will-change: transform para optimización del navegador
- Eliminación de transformaciones 3D conflictivas
- Simplificación de la jerarquía de animaciones
- Uso de !important estratégico en estados hover

3. ¿Qué técnica se usó para garantizar 60fps en la animación orbital?

Solución:

- Implementación de requestAnimationFrame
- Cálculo basado en deltaTime
- Eliminación de forced reflows
- **Optimización con transform en lugar de propiedades costosas**

3. ¿Cómo se logró la adaptabilidad responsive?



Solución:

- Unidades relativas (vh, %)
- Media queries con breakpoints
- Rediseño progresivo de elementos
- Simplificación de efectos en móviles

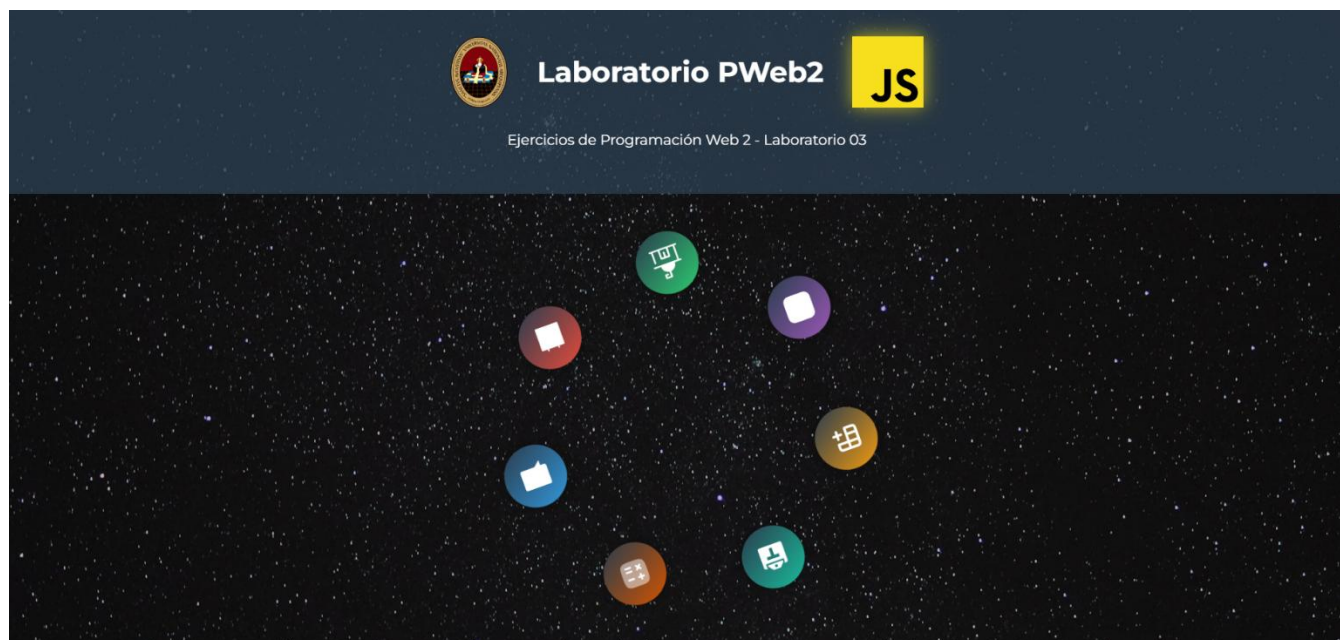
4. ¿Qué patrón se siguió para la organización del código JavaScript?

Solución:

- Separación por responsabilidades (orbital vs ejercicios)
- IIFE para evitar contaminación global en el código
- Delegación de eventos
- Cacheo de selectores DOM

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p align="center">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p align="center">Código: GUIA-PRLE-001</p>	<p align="right">Página: 8</p>

(Algunas capturas sobre el buen funcionamiento de nuestra página web)



« Volver

Editor de Texto

Cambia tamaño y color de un texto

Código
Ejecutar

Simulación

```

let tamañoActual = 16;

function cambiarTamaño(accion) {
  const texto = document.getElementById('textoEditable');
  if (accion === 'aumentar') {
    tamañoActual += 2;
  } else if (accion === 'disminuir' && tamañoActual > 10)
    tamañoActual -= 2;
  }
  texto.style.fontSize = `${tamañoActual}px`;
}

function cambiarColor(color) {
  const texto = document.getElementById('textoEditable');
  const colores = {
    rojo: '#ff0000',
    azul: '#0000ff',
    verde: '#00ff00',
    negro: '#000000'
  };
  texto.style.color = colores[color] || colores.negro;
}

```

Este es un texto de ejemplo que puedes modificar.

Tamaño:

Aumentar
Disminuir

Color:

Rojo
Azul
Verde
Negro

Consola

> Tamaño actual: 18px

> Tamaño actual: 20px

« Volver

Suma de Tabla Dinámica

Genera una tabla aleatoria y calcula la suma

Código
Ejecutar

Simulación

```

function generarTabla(filas) {
  const tabla = [];
  for (let i = 0; i < filas; i++) {
    tabla.push(Math.floor(Math.random() * 100) + 1);
  }
  return tabla;
}

function calcularSuma(tabla) {
  return tabla.reduce((sum, num) => sum + num, 0);
}

```

Generar Tabla
Calcular Suma

Tabla:

Índice	Valor
1	34
2	55
3	81

Suma:

170



Consola

> Tabla generada:

> Suma calculada: 760

> Tabla generada:

> Suma calculada: 170

	<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;">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: 10</p>

Lecciones Aprendidas y Retroalimentación

- **Optimización de animaciones:** El uso excesivo de transformaciones 3D afecta el rendimiento
- **Manejo de tiempo:** requestAnimationFrame supera a setTimeout para animaciones
- **Responsive design:** Las unidades viewport (vh/vw) son esenciales
- **Accesibilidad:** Contraste de colores y semántica HTML mejoran la usabilidad

Recomendaciones para Futuros Laboratorios

1. **Implementar Web Workers para cálculos intensivos**
2. **Añadir modo oscuro/light con CSS variables**
3. **Integrar API de reconocimiento de voz para interacciones**
4. **Implementar sistema de persistencia con localStorage**



CONCLUSIONES

Colocar las conclusiones, apreciaciones reflexivas, opiniones finales a cerca de los resultados obtenidos de la sesión de laboratorio.

El desarrollo de los ejercicios implementa técnicas fundamentales de JavaScript como manipulación de strings (para invertir texto), manejo de objetos Date (en cálculo de días), operaciones con arrays (en la tabla dinámica) y manipulación del DOM (en el editor de texto). Estas funcionalidades, aunque aparentemente simples en su concepción individual, conforman en conjunto una base sólida que demuestra el dominio de los conceptos clave de programación web.

La arquitectura del sistema utiliza un enfoque modular que separa claramente:

1. La presentación visual (animaciones CSS y diseño responsive).
2. La lógica de interacción (sistema orbital en JavaScript).
3. Los ejercicios independientes (cada uno con su propia implementación).

	<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;">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: 11</p>

METODOLOGÍA DE TRABAJO

Colocar la metodología de trabajo que ha utilizado el estudiante o el grupo para resolver la práctica, es decir el procedimiento/secuencia de pasos en forma general.

- a) **Comprensión del problema:** En esta etapa, revisé cada una de las actividades propuestas, identificando cuidadosamente las restricciones y los objetivos a alcanzar.
- b) **Diseño del algoritmo:** Planifiqué la secuencia lógica necesaria para implementar la solución, aplicando los conocimientos adquiridos en Programación Web I y II.
- c) **Codificación:** Procedí a implementar los programas solicitados, asegurándome de utilizar correctamente los arreglos y métodos.
- d) **Pruebas:** Realicé pruebas adicionales para verificar que el código funcionara de manera correcta con diferentes casos de prueba.

REFERENCIAS Y BIBLIOGRAFÍA

Colocare las referencias utilizadas para el desarrollo de la práctica en formato IEEE

- Javascript tutorial. \url <https://www.w3schools.com/js/>, 2024. Accessed: 02-05-2024.
- Loiane Groner. *Learning JavaScript Data Structures and Algorithms: Write complex and powerful*
- <https://github.com/Q3son/Pweb2---Edson-Subia.git>
- https://youtu.be/U8r_r-8fS2Y