

 ESPE UNIVERSIDAD DE LAS FUERZAS ARMADAS INNOVACIÓN PARA LA EXCELENCIA	INFORME DE TAREA	VERSIÓN: 1.0 FECHA ÚLTIMA REVISIÓN: 18/06/2024
--	-------------------------	--

DEPARTAMENTO	CIENCIAS DE LA COMPUTACIÓN	CARRERA	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN		
ASIGNATURA	PROGRAMACIÓN INTEGRATIVA	PERIODO LECTIVO	MAY - SEP 24	NIVEL	7MO
DOCENTE	ING. JOSE ALBERTO SANCHO, MSC.	NRC	16496	TAREA N°	6
TEMA DE LA PRÁCTICA	APLICACIONES PARA WEB COMPONENTS.				

INTRODUCCIÓN:

En el panorama actual del desarrollo web, la adopción de tecnologías modernas es clave para crear aplicaciones web eficientes, escalables y atractivas. Una de estas tecnologías destacadas son los Web Components, un estándar de la web que permite encapsular funcionalidades y estilos en componentes reutilizables. Estos componentes ofrecen una forma modular de construir interfaces de usuario, lo que facilita la creación y mantenimiento de aplicaciones web complejas.

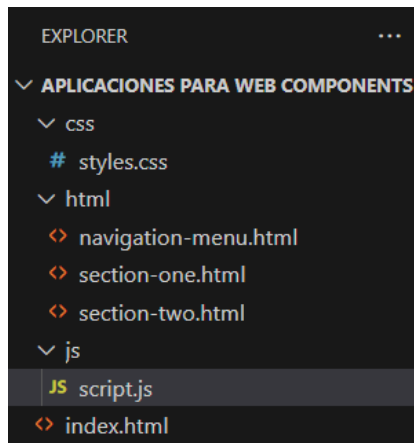
Los Web Components se basan en estándares abiertos de la web, como el Shadow DOM y las Custom Elements, lo que los hace altamente interoperables y compatibles con diferentes frameworks y bibliotecas. Esta capacidad de interoperabilidad y reutilización es fundamental en el desarrollo moderno, ya que permite a los desarrolladores crear aplicaciones web más flexibles y adaptables a medida que evolucionan las necesidades del mercado y las tecnologías emergentes.

DESARROLLO:

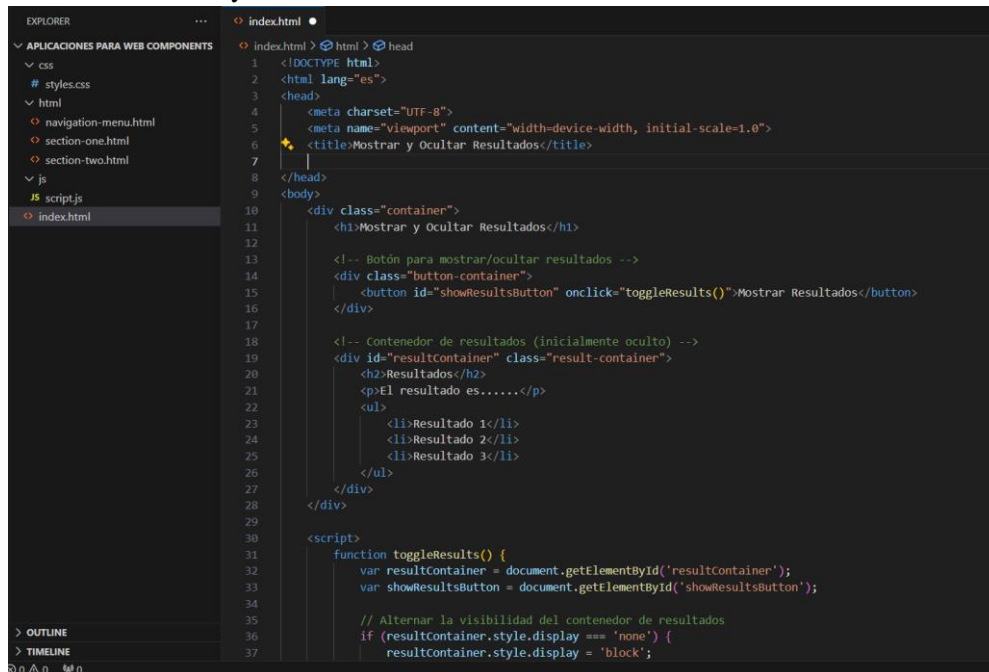
- Herramientas necesarias, para este caso se ocupó Visual Studio Code y un navegador web.



- Se crea una carpeta en la cual se va a encontrar de manera organizada por carpetas (css, js, html y un archivo html como página principal).



3. Dentro de nuestro archivo index.html (Principal) se proce a visualizar los pasos para crear el esqueleto HTML básico etiquetas <html>, <head>, <body>. La inclusión de elementos como encabezados, párrafos, etc y enlaces del archivo CSS y JS al HTML.

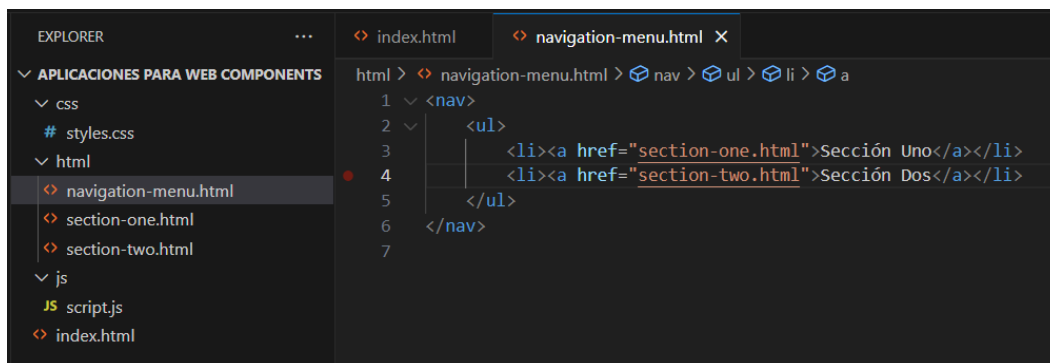


```

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>Mostrar y Ocultar Resultados</title>
7 </head>
8 <body>
9   <div class="container">
10    <h1>Mostrar y Ocultar Resultados</h1>
11
12    <!-- Botón para mostrar/ocultar resultados -->
13    <div class="button-container">
14      <button id="showResultsButton" onclick="toggleResults()">Mostrar Resultados</button>
15    </div>
16
17    <!-- Contenedor de resultados (inicialmente oculto) -->
18    <div id="resultContainer" class="result-container">
19      <h2>Resultados</h2>
20      <p>El resultado es.....</p>
21      <ul>
22        <li>Resultado 1</li>
23        <li>Resultado 2</li>
24        <li>Resultado 3</li>
25      </ul>
26    </div>
27  </div>
28 </body>
29 </html>
30
31 <script>
32   function toggleResults() {
33     var resultContainer = document.getElementById('resultContainer');
34     var showResultsButton = document.getElementById('showResultsButton');
35
36     // Alternar la visibilidad del contenedor de resultados
37     if (resultContainer.style.display === 'none') {
38       resultContainer.style.display = 'block';
39     }
34
35
36
37

```

4. Dentro de nuestro archivo ../html/navigarion-menu.html (Secundario) se proce a visualizar etiquetas <nav>

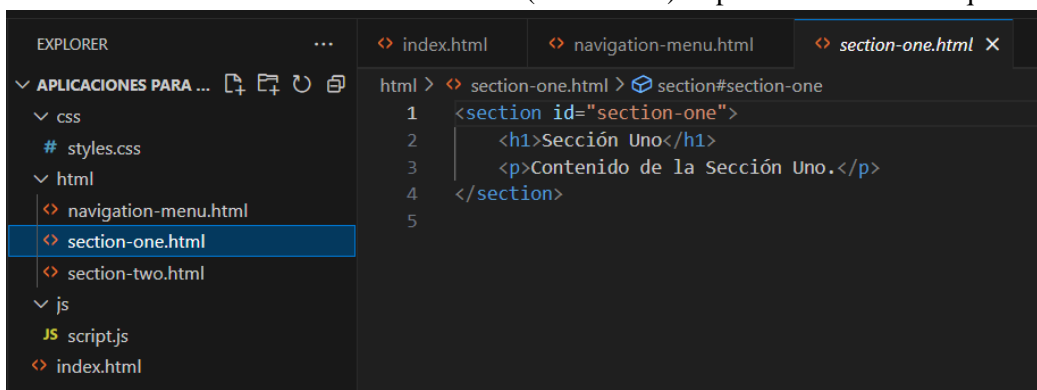


```

1 <nav>
2   <ul>
3     <li><a href="section-one.html">Sección Uno</a></li>
4     <li><a href="section-two.html">Sección Dos</a></li>
5   </ul>
6 </nav>
7

```

5. Dentro de nuestro archivo ../html/section-one.html (Secundario) se proce a visualizar etiquetas <nav>.

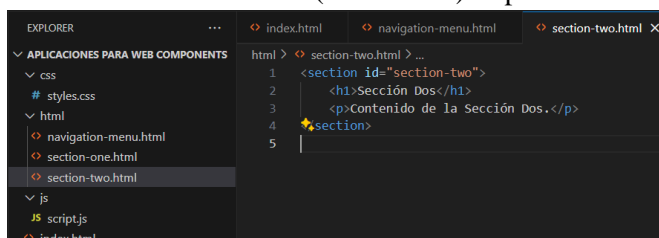


```

1 <section id="section-one">
2   <h1>Sección Uno</h1>
3   <p>Contenido de la Sección Uno.</p>
4 </section>
5

```

6. Dentro de nuestro archivo `../html/section-two.html` (Secundario) se proce a visualizar etiquetas `<nav>`.

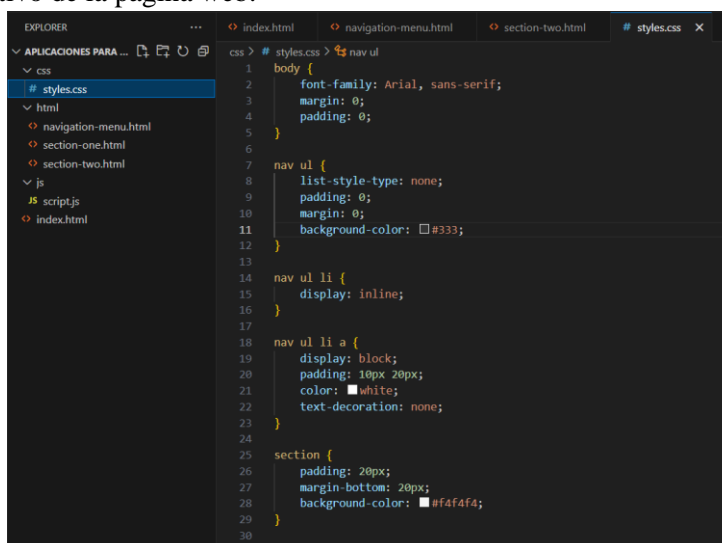


```

1 <section id="section-two">
2   <h1>Sección Dos</h1>
3   <p>Contenido de la Sección Dos.</p>
4   <section>
5

```

7. Dentro del archivo `styles.css` se encuentra estilización con CSS y se visualiza el uso de selectores CSS para estilizar elementos HTML. La aplicación de propiedades como color de fondo, tamaño de fuente, márgenes, rellenos y diseño creativo de la página web.

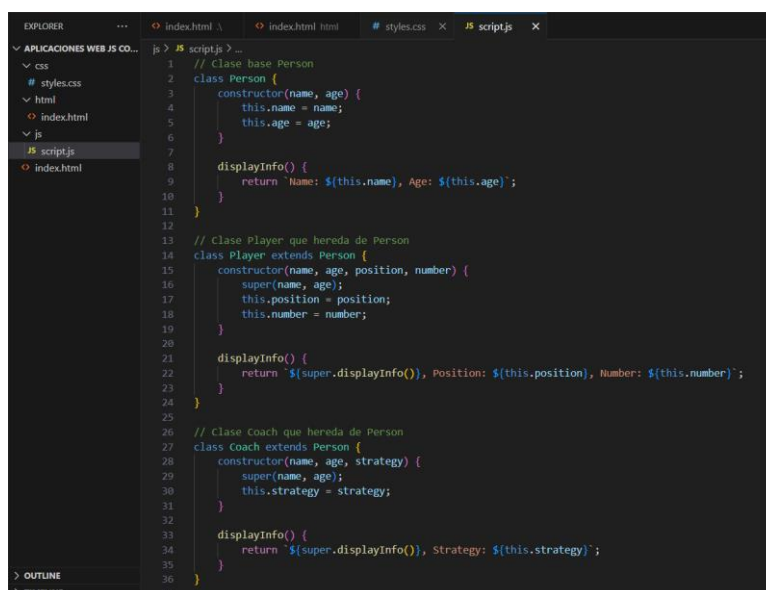


```

1 body {
2   font-family: Arial, sans-serif;
3   margin: 0;
4   padding: 0;
5 }
6
7 nav ul {
8   list-style-type: none;
9   padding: 0;
10  margin: 0;
11  background-color: #333;
12 }
13
14 nav ul li {
15   display: inline;
16 }
17
18 nav ul li a {
19   display: block;
20   padding: 10px 20px;
21   color: white;
22   text-decoration: none;
23 }
24
25 section {
26   padding: 20px;
27   margin-bottom: 20px;
28   background-color: #f4f4f4;
29 }
30

```

8. En la carpeta `js`, crea un archivo `script.js` para definir el comportamiento dinámico de tus componentes utilizando JavaScript. Implementa la lógica de los componentes utilizando la API de Web Components, como la definición de Custom Elements, la gestión del ciclo de vida y la manipulación del DOM según sea necesario.

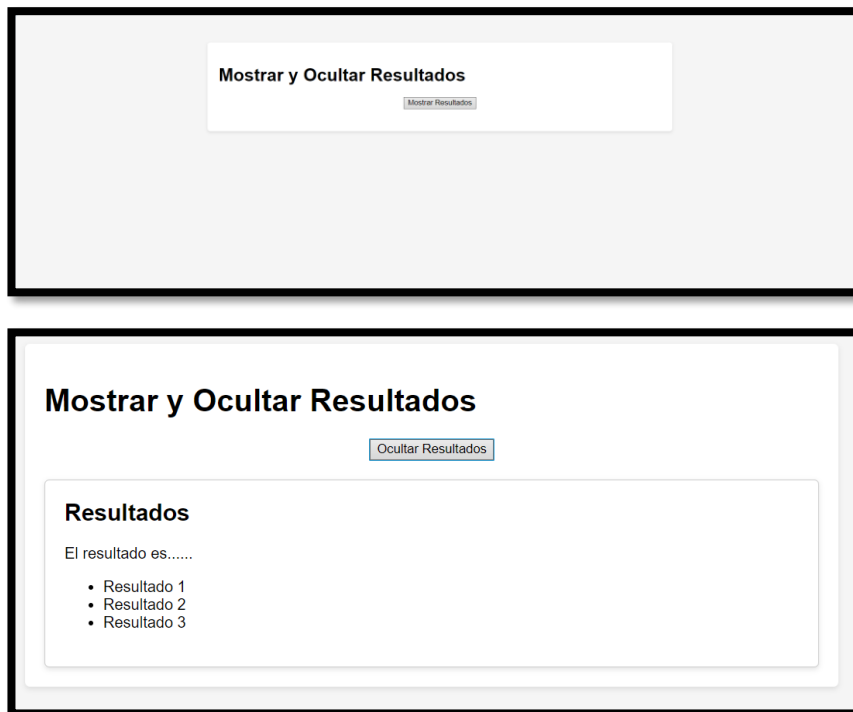


```

1 // Clase base Person
2 class Person {
3   constructor(name, age) {
4     this.name = name;
5     this.age = age;
6   }
7
8   displayInfo() {
9     return `Name: ${this.name}, Age: ${this.age}`;
10  }
11 }
12
13 // Clase Player que hereda de Person
14 class Player extends Person {
15   constructor(name, age, position, number) {
16     super(name, age);
17     this.position = position;
18     this.number = number;
19   }
20
21   displayInfo() {
22     return `${super.displayInfo()}, Position: ${this.position}, Number: ${this.number}`;
23   }
24 }
25
26 // Clase Coach que hereda de Person
27 class Coach extends Person {
28   constructor(name, age, strategy) {
29     super(name, age);
30     this.strategy = strategy;
31   }
32
33   displayInfo() {
34     return `${super.displayInfo()}, Strategy: ${this.strategy}`;
35   }
36 }
37

```

9. Se entra al paso de pruebas y ajustes y para eso probamos la aplicación en nuestro navegador.



Mostrar y Ocultar Resultados

Mostrar Resultados

Mostrar y Ocultar Resultados

Ocultar Resultados

Resultados

El resultado es.....

- Resultado 1
- Resultado 2
- Resultado 3

IMPORTANCIA

La práctica y el tema de las aplicaciones basadas en Web Components son de suma importancia en el desarrollo web moderno debido a su capacidad para crear interfaces de usuario modulares, reutilizables y fácilmente mantenibles. Al utilizar Web Components, los desarrolladores pueden encapsular funcionalidades complejas en componentes independientes, lo que facilita la construcción de aplicaciones escalables y flexibles. Además, al seguir los estándares de la web y aprovechar las ventajas del Shadow DOM y las Custom Elements, las aplicaciones basadas en Web Components ofrecen una mejor organización del código, mejorando la legibilidad, la reusabilidad y la interoperabilidad del software desarrollado.

CONCLUSIONES:

- El uso de Web Components permite desarrollar aplicaciones web con componentes flexibles y altamente reutilizables, lo que agiliza el proceso de desarrollo y facilita la mantenibilidad del código.
- Al seguir estándares abiertos y aprovechar las capacidades del Shadow DOM y las Custom Elements, las aplicaciones basadas en Web Components son interoperables entre diferentes frameworks y bibliotecas, fomentando un ecosistema web más colaborativo y compatible.
- La estructura modular de Web Components mejora la organización y legibilidad del código, lo que facilita la colaboración entre equipos de desarrollo y la escalabilidad de las aplicaciones a medida que crecen en complejidad.

ELABORADO POR:

CUADRADO AVILÉS SANTIAGO ALEJANDRO
ESTUDIANTE