

 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°	5
TEMA DE LA PRÁCTICA	APLICACIONES WEB JS CON POO.				

INTRODUCCIÓN:

En la actualidad, la Programación Orientada a Objetos (OOP, por sus siglas en inglés) se ha convertido en una de las metodologías más utilizadas en el desarrollo de software debido a su capacidad para modelar sistemas complejos de manera intuitiva y organizada. Este paradigma se basa en la creación de objetos que representan entidades del mundo real, permitiendo la reutilización de código, la encapsulación de datos y la creación de relaciones entre los diferentes componentes de un programa. La implementación de OOP en JavaScript ha facilitado la construcción de aplicaciones web dinámicas y robustas, ofreciendo una forma estructurada y modular de escribir código.

En este proyecto, se ha desarrollado una aplicación web con la temática de fútbol utilizando los principios de la Programación Orientada a Objetos en JavaScript. La aplicación gestiona la información de un equipo de fútbol, incluyendo detalles sobre los jugadores, el entrenador y el estadio. A través de este trabajo, se ha demostrado cómo utilizar las relaciones de generalización, asociación, agregación y composición para crear un modelo de clases que representa un equipo de fútbol. El objetivo es aplicar los fundamentos y principios de OOP para resolver problemas del mundo real, mostrando la flexibilidad y potencia de JavaScript en el desarrollo de aplicaciones web.

DESARROLLO:

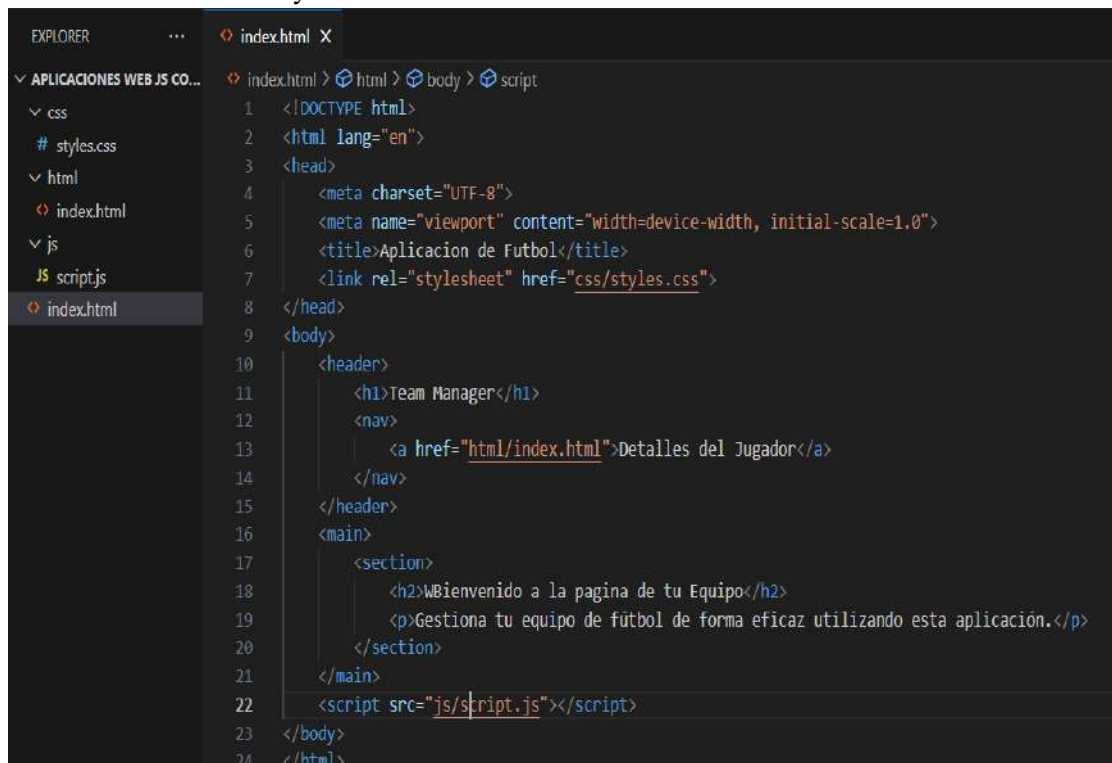
1. Herramientas necesarias, para este caso se ocupó Visual Studio Code y un navegador web.



2. 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).

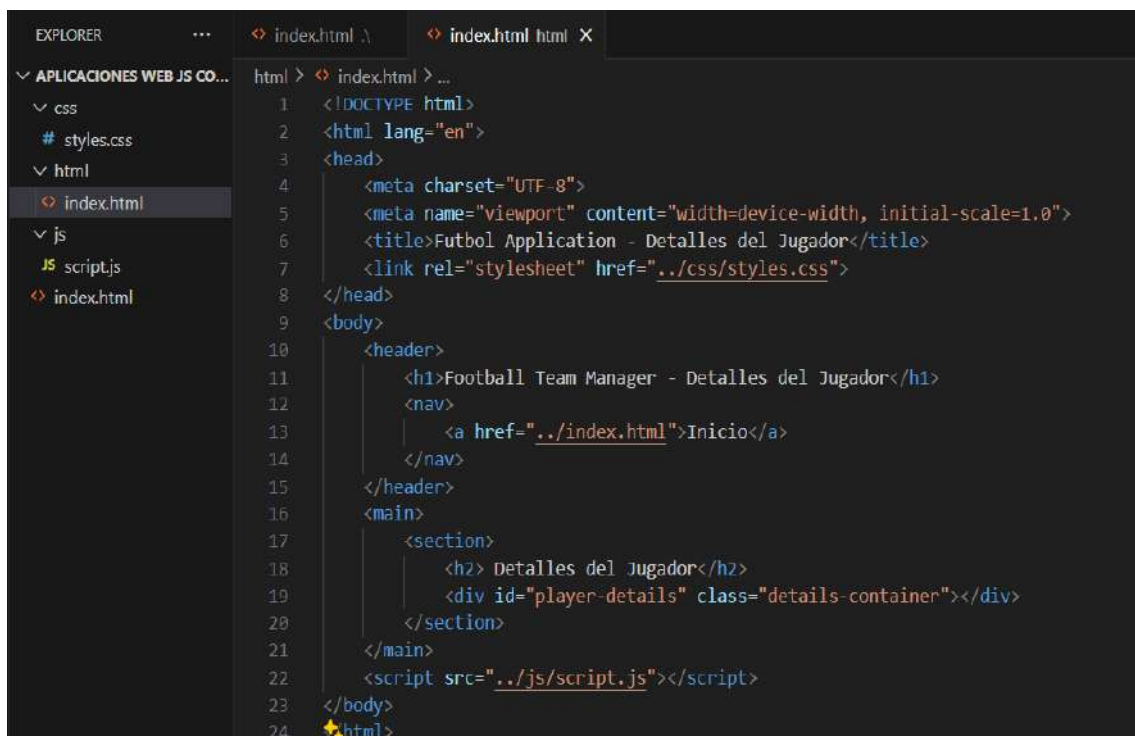


3. Dentro de nuestro archivo index.html (Principal) se proce a visualizar los pasos para crear el esqueleto HTML básico etiquetas <html>, <head>, <body>. La inclusión de elementos como encabezados, párrafos, etc y enlaces del archivo CSS y JS al HTML.



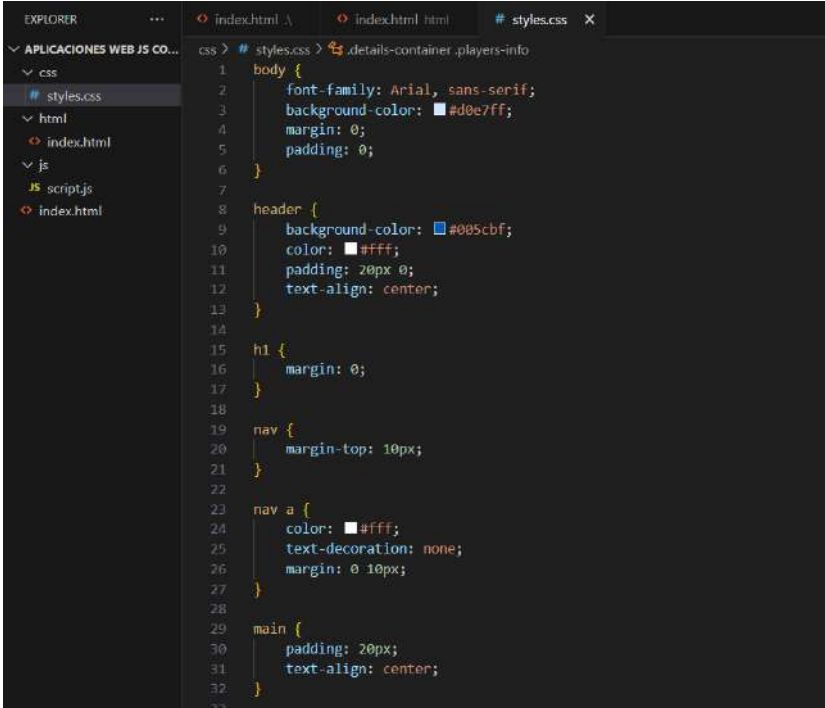
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Aplicacion de Futbol</title>
7   <link rel="stylesheet" href="css/styles.css">
8 </head>
9 <body>
10   <header>
11     <h1>Team Manager</h1>
12     <nav>
13       <a href="html/index.html">Detalles del Jugador</a>
14     </nav>
15   </header>
16   <main>
17     <section>
18       <h2>¡Bienvenido a la pagina de tu Equipo!</h2>
19       <p>Gestiona tu equipo de fútbol de forma eficaz utilizando esta aplicación.</p>
20     </section>
21   </main>
22   <script src="js/script.js"></script>
23 </body>
24 </html>
```

4. Dentro de nuestro archivo ../html/index.html (Secundario) se proce a visualizar etiquetas <html>, <head>, <body>. La inclusión de elementos como encabezados, párrafos, etc y enlaces del archivo CSS y JS al HTML.



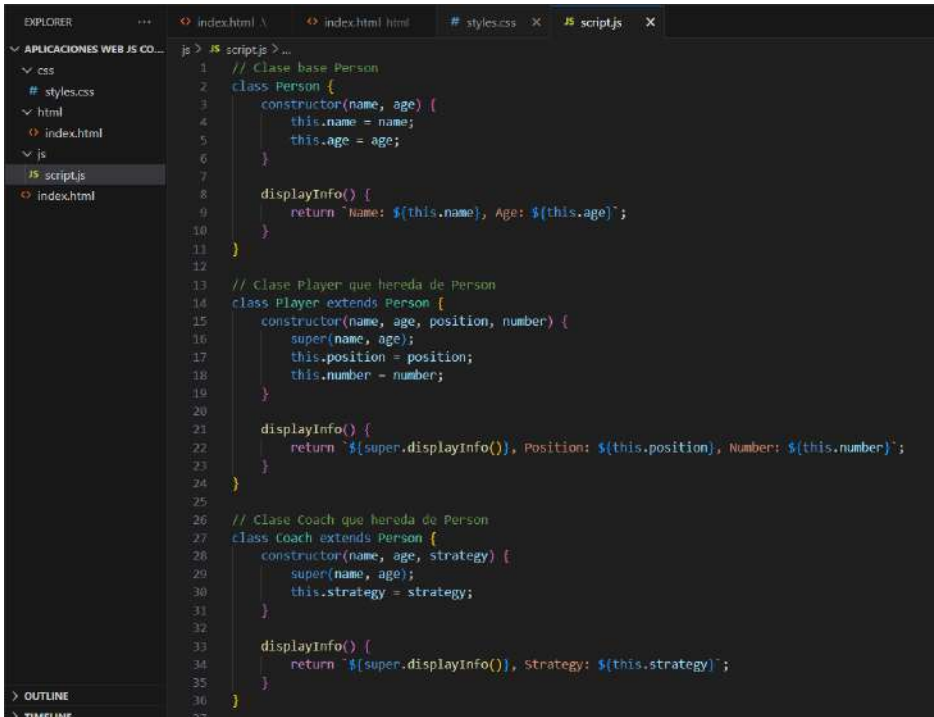
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Futbol Application - Detalles del Jugador</title>
7   <link rel="stylesheet" href="../css/styles.css">
8 </head>
9 <body>
10   <header>
11     <h1>Football Team Manager - Detalles del Jugador</h1>
12     <nav>
13       <a href="../index.html">Inicio</a>
14     </nav>
15   </header>
16   <main>
17     <section>
18       <h2>Detalles del Jugador</h2>
19       <div id="player-details" class="details-container"></div>
20     </section>
21   </main>
22   <script src="../js/script.js"></script>
23 </body>
24 </html>
```

5. 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.



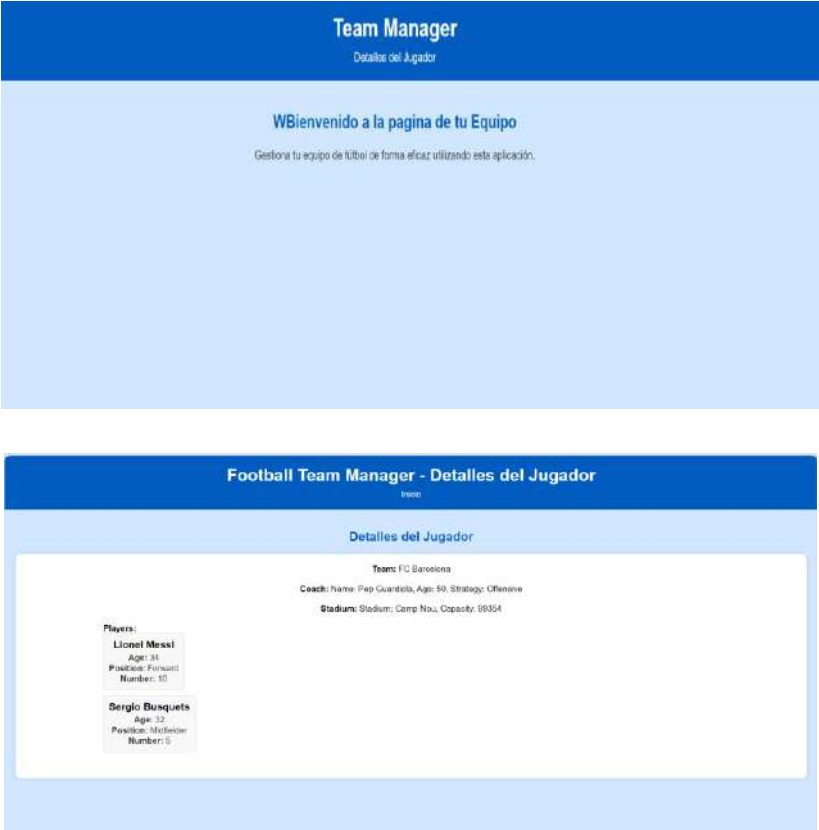
```
css > # styles.css > .details-container.players-info
1 body {
2   font-family: Arial, sans-serif;
3   background-color: #d0e7ff;
4   margin: 0;
5   padding: 0;
6 }
7
8 header {
9   background-color: #005cbf;
10  color: #fff;
11  padding: 20px 0;
12  text-align: center;
13 }
14
15 h1 {
16   margin: 0;
17 }
18
19 nav {
20   margin-top: 10px;
21 }
22
23 nav a {
24   color: #fff;
25   text-decoration: none;
26   margin: 0 10px;
27 }
28
29 main {
30   padding: 20px;
31   text-align: center;
32 }
33
```

6. El archivo script.js es fundamental para la implementación de la lógica y la estructura de nuestra aplicación web basada en el paradigma de Programación Orientada a Objetos (OOP). En este archivo, se definen las clases que representan los diferentes componentes del equipo de fútbol, así como las relaciones entre ellas.



```
js > JS script.js > ...
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
```

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



IMPORTANCIA

La Programación Orientada a Objetos (OOP) es esencial en el desarrollo de software debido a su capacidad para organizar el código de manera modular y reutilizable. Al centrarse en la creación de objetos que representan entidades del mundo real, la OOP facilita la comprensión y el mantenimiento del código, promoviendo la colaboración en equipos de desarrollo y mejorando la seguridad mediante la encapsulación de datos. Además, la herencia y las relaciones de agregación/composición en OOP permiten la reutilización eficiente de código y la modelación de estructuras complejas, proporcionando un enfoque estructurado y claro para diseñar sistemas informáticos.

- CONCLUSIONES:**
- La Programación Orientada a Objetos (OOP) mejora la organización del código al centrarse en la creación de objetos que representan entidades del mundo real, lo que facilita su comprensión y mantenimiento.
 - La OOP promueve la reutilización de código a través de la herencia y las relaciones de agregación/composición, lo que permite desarrollar sistemas informáticos de manera eficiente y escalable.
 - La encapsulación en la OOP aumenta la seguridad al limitar el acceso a las propiedades y métodos de los objetos, lo que reduce los errores y mejora la calidad del software desarrollado.

ELABORADO POR:

CUADRADO AVILÉS SANTIAGO ALEJANDRO

ESTUDIANTE