

Universidad Tecnológica Metropolitana

Aplicaciones Web

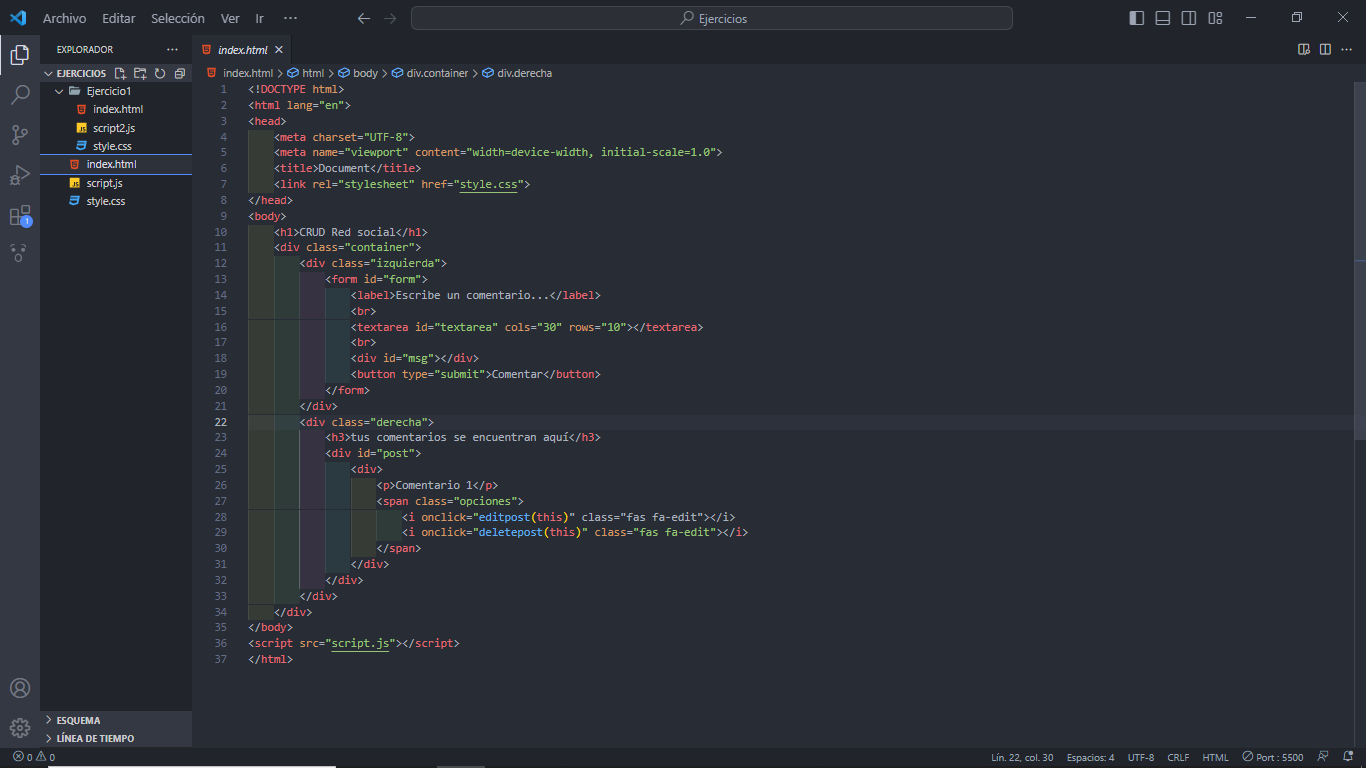
Desarrollo de Software Multiplataforma

Canul Mena Gustavo Alejandro

Diego May

3B

Introducción a DOM de Javascript



1. `<!DOCTYPE html>`: Esta declaración define el tipo de documento como HTML5.

2. `<html lang="en">`: El elemento `<html>` es el contenedor raíz de todo el documento HTML. El atributo `lang` establece el idioma predeterminado de la página.

3. `<head>`: La sección `<head>` contiene información sobre el documento, como metadatos y enlaces a archivos externos.

4. `<meta charset="UTF-8">`: Este meta elemento especifica la codificación de caracteres del documento como UTF-8, que admite una amplia gama de caracteres.

5. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Este meta elemento configura la visualización de la página en dispositivos móviles ajustando el ancho a la anchura del dispositivo y estableciendo una escala inicial de 1.0.

6. `<title>Document</title>`: El elemento `<title>` define el título de la página, que se muestra en la pestaña del navegador o en los resultados de búsqueda.

7. `<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" integrity="sha512-iecdLmaskl7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcenmpD1ypASozpmT/E0iPtmFIB46ZmdtAc9eNBvH0H/ZpiBw==" crossorigin="anonymous" referrerpolicy="no-referrer" />`: Este enlace `<link>` carga una hoja de estilos CSS de Font Awesome, que proporciona iconos para su uso en la página.

8. `<link rel="stylesheet" href="style.css">`: Este enlace `<link>` carga una hoja de estilos CSS desde un archivo llamado "style.css" ubicado en el mismo directorio que el archivo HTML actual.

9. `<body>`: El elemento `<body>` contiene todo el contenido visible de la página.

10. `<h1>CRUD Red social</h1>`: El encabezado `<h1>` muestra el título principal de la página.

11. `<div class="container">`: El `<div>` con la clase "container" crea un contenedor para los elementos principales de la página.

12. `<div class="left">`: El `<div>` con la clase "left" define una sección para el formulario de comentarios.

13. `<form id="form">`: El elemento `<form>` representa un formulario en la página y tiene un atributo `id` establecido en "form" para identificarlo de manera única.

14. `<label>Escribe un comentario...</label>`: La etiqueta `<label>` proporciona una descripción o etiqueta para el campo de entrada de texto siguiente.

15. `<textarea id="textarea" cols="30" rows="10"></textarea>`: El elemento `<textarea>` crea un área de texto multilínea donde los usuarios pueden escribir sus comentarios. Tiene un atributo `id` establecido en "textarea" para identificarlo de manera única.

16. `<div id="msg"></div>`: El `<div>` con el `id` "

msg" se utilizará para mostrar mensajes o notificaciones relacionadas con el envío del formulario.

17. `<button type="submit">Comentar</button>`: El botón `<button>` se utiliza para enviar el formulario cuando se hace clic en él.

18. `</form>`: Cierra el formulario.

19. `</div>`: Cierra el `<div>` con la clase "left".

20. `<div class="right">`: El `<div>` con la clase "right" define una sección para mostrar los comentarios existentes.

21. `<h3>Tus comentarios estan aqui</h3>`: El encabezado `<h3>` muestra un título para la sección de comentarios.

22. `<div id="post">`: El `<div>` con el `id` "post" es un contenedor para mostrar los comentarios existentes.

23. `<div>`: Crea un contenedor para un comentario individual.

24. `<p>Comentario 1</p>`: El elemento `<p>` muestra el contenido de un comentario.

25. `<span class="options">`: El `<span>` con la clase "options" crea un contenedor para las opciones de edición y eliminación de un comentario.

26. `<i onclick="editpost(this)" class="fas fa-edit"></i>`: El elemento `<i>` es un ícono de Font Awesome que se muestra como un ícono de edición. El atributo `onclick` llama a la función "editpost" en JavaScript cuando se hace clic en el ícono.

27. `<i onclick="deletepost(this)" class="fas fa-trash-alt"></i>`: El elemento `<i>` es otro ícono de Font Awesome que se muestra como un ícono de eliminación. El atributo `onclick` llama a la función "deletepost" en JavaScript cuando se hace clic en el ícono.

28. `</span>`: Cierra el `<span>` con la clase "options".

29. `</div>`: Cierra el contenedor para un comentario individual.

30. `</div>`: Cierra el `<div>` con el `id` "post".

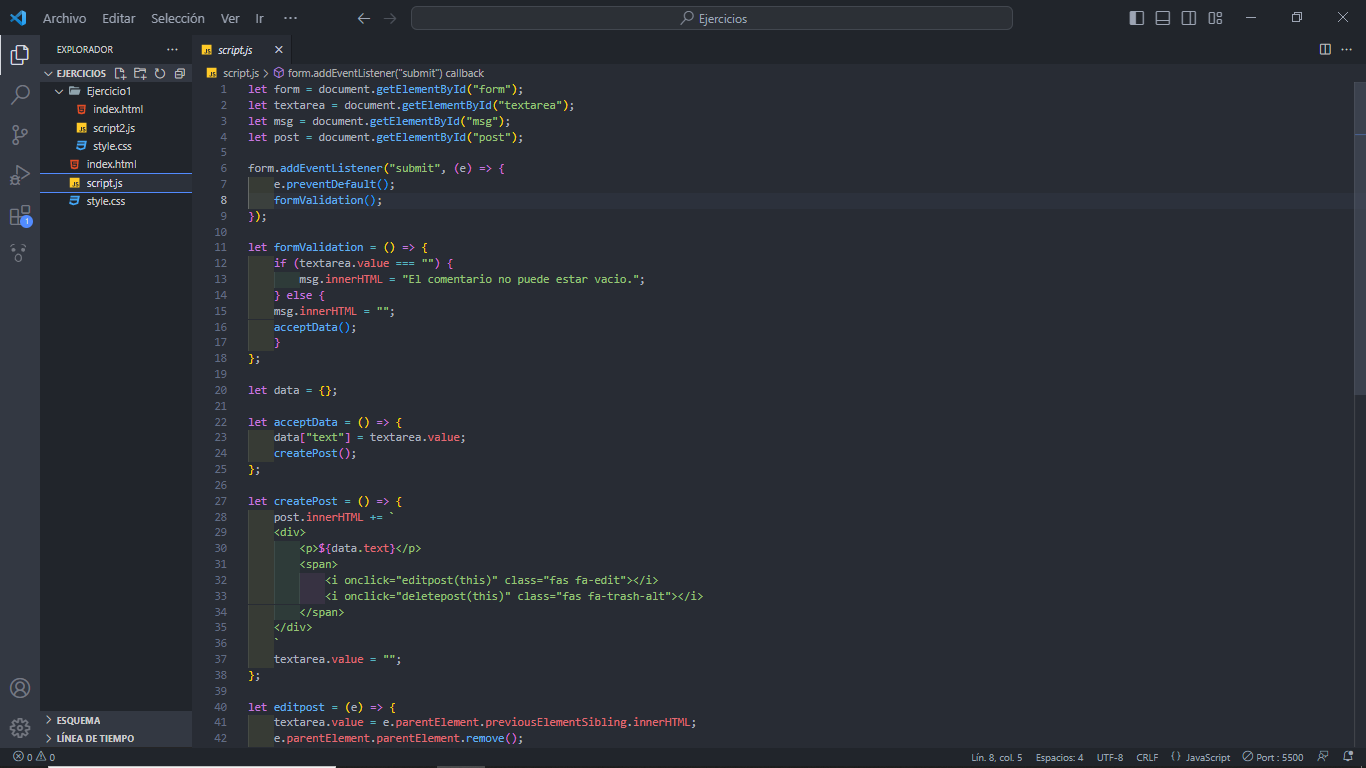
31. `</div>`: Cierra el `<div>` con la clase "right".

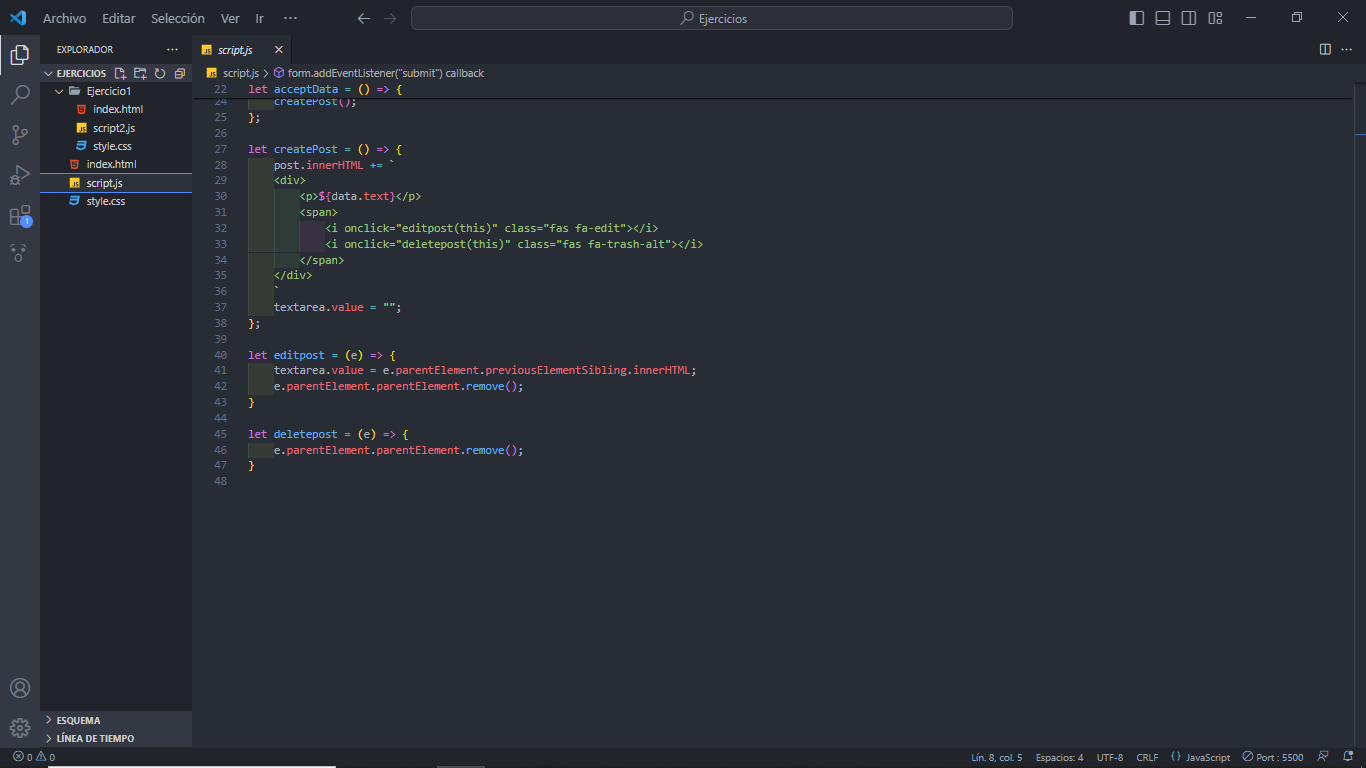
32. `</div>`: Cierra el `<div>` con la clase "container".

33. `</body>`: Cierra el `<body>`.

34. `<script src="script.js"></script>`: El elemento `<script>` se utiliza para cargar un archivo de script externo llamado "script.js" que contiene código JavaScript para la interacción y funcionalidad dinámica de la página.

35. `</html>`: Cierra el elemento `<html>`.





1. `let form = document.getElementById("form");`: Esta línea de código obtiene una referencia al elemento del formulario HTML con el `id` "form" utilizando `getElementById`.

2. `let textarea = document.getElementById("textarea");`: Esta línea obtiene una referencia al elemento de entrada de texto de comentarios HTML con el `id` "textarea" utilizando `getElementById`.

3. `let msg = document.getElementById("msg");`: Esta línea obtiene una referencia al elemento `<div>` con el `id` "msg" que se utiliza para mostrar mensajes o notificaciones al usuario.

4. `let post = document.getElementById("post");`: Esta línea obtiene una referencia al elemento `<div>` con el `id` "post" que se utiliza para mostrar los comentarios existentes.

5. `form.addEventListener("submit", (e) => {...});`: Este código agrega un evento "submit" al formulario. Cuando se envía el formulario, se ejecuta la función de flecha proporcionada.

6. `e.preventDefault();`: Esta línea de código evita el comportamiento predeterminado del formulario al enviarlo, lo que significa que la página no se recargará.

7. `let formValidation = () => {...};`: Esta función realiza la validación del formulario antes de enviar los datos. Verifica si el campo de texto de comentarios está vacío o no.

8. `if (textarea.value === "") {...};`: Esta condición verifica si el campo de texto de comentarios está vacío.

9. `msg.innerHTML = "El comentario no puede estar vacio.";`: Si el campo de texto está vacío, se asigna el mensaje de error al elemento `<div>` con el `id` "msg" utilizando la propiedad `innerHTML`.

10. `msg.innerHTML = "";`: Si el campo de texto no está vacío, se vacía el mensaje de error estableciendo `innerHTML` en una cadena vacía.

11. `let data = {};`: Se declara un objeto vacío llamado "data" para almacenar los datos del comentario.

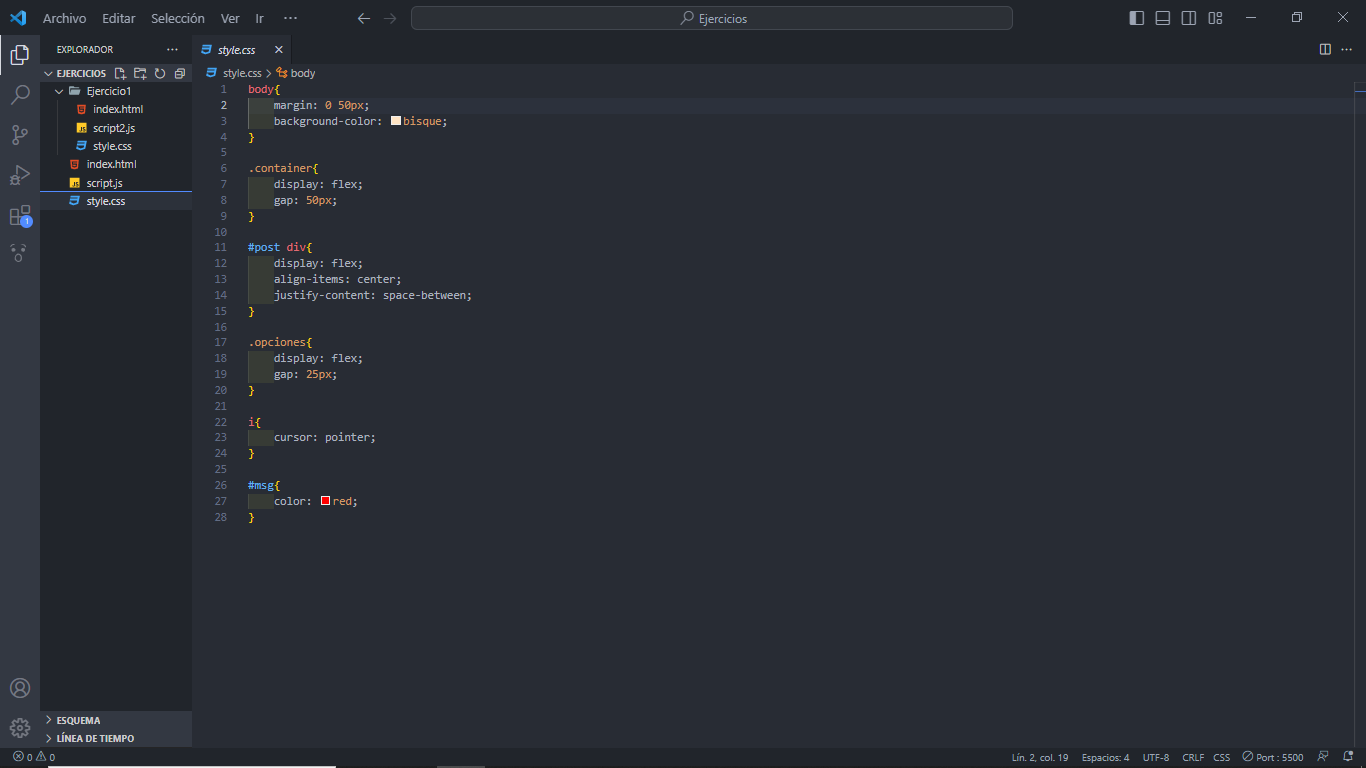
12. `let acceptData = () => {...};`: Esta función guarda el valor del campo de texto de comentarios en la propiedad "text" del objeto "data".

13. `let createPost = () => {...};`: Esta función se encarga de crear el HTML para mostrar un nuevo comentario en la sección de comentarios existentes. Agrega el contenido del comentario a la estructura del HTML en el elemento `<div>` con el `id` "post" utilizando la propiedad `innerHTML`. También se restablece el campo de texto de comentarios a un valor vacío.

14. `let editpost = (e) => {...};`: Esta función se llama cuando se hace clic en el ícono de edición de un comentario existente. Establece el valor del campo de texto de comentarios al contenido del comentario seleccionado para permitir la edición. Luego, elimina el contenedor del comentario existente del DOM.

15. `let deletepost = (e) => {...};`: Esta función se llama cuando se hace clic en el ícono de eliminación de un comentario existente. Elimina el contenedor del comentario seleccionado del DOM.

En general, el código JavaScript se encarga de la validación del formulario, la captura de datos del campo de texto de comentarios, la creación y eliminación de comentarios, y la manipulación del DOM para reflejar los cambios realizados. Esta implementación básica de CRUD (Crear, Leer, Actualizar, Eliminar) permite al usuario agregar comentarios, editarlos y eliminarlos.



1. `body{ margin: 0 50px; }`: Esta regla establece un margen de 0 píxeles en la parte superior e inferior del cuerpo de la página y un margen de 50 píxeles en los lados izquierdo y derecho.

2. `.container{ display: flex; gap:50px; }`: Esta regla aplica un contenedor flexible a los elementos con la clase "container", lo que permite que los elementos internos se muestren en una fila. El espacio entre los elementos se establece en 50 píxeles utilizando la propiedad `gap`.

3. `#post{ width: 400px; }`: Esta regla establece un ancho de 400 píxeles para el elemento con el `id` "post". Esto controla el ancho de la sección donde se muestran los comentarios existentes.

4. `#post div{ display: flex; align-items: center; justify-content: space-between; }`: Esta regla aplica un contenedor flexible a los elementos `<div>` dentro de "#post". Los elementos internos se alinean verticalmente y se distribuyen uniformemente en el espacio disponible.

5. `.options{ display: flex; gap: 25px; }`: Esta regla aplica un contenedor flexible a los elementos con la clase "options". Los elementos internos se muestran en una fila con un espacio de 25 píxeles entre ellos.

6. `i{ cursor: pointer; }`: Esta regla establece el cursor del mouse como "pointer" (un ícono de mano) cuando se pasa sobre los elementos `<i>`. Esto indica que los elementos `<i>` son interactivos y se pueden hacer clic en ellos.

7. `#msg{ color: red; }`: Esta regla establece el color del texto del elemento con el `id` "msg" como rojo. Esto se utiliza para mostrar mensajes de error en rojo.

Conclusiones

El código HTML establece una estructura básica de una página web que contiene un formulario de comentarios y una sección para mostrar los comentarios existentes. El código JavaScript maneja la validación del formulario, la captura de datos del campo de texto de comentarios, la creación de nuevos comentarios y la funcionalidad de edición y eliminación de comentarios existentes. Las reglas CSS proporcionadas ayudan a dar estilo a los elementos de la página, como el diseño de la sección de comentarios, el espaciado entre elementos y la apariencia de los íconos utilizados para editar y eliminar comentarios. En general, el código proporcionado crea una funcionalidad básica de una red social donde los usuarios pueden agregar comentarios, editarlos y eliminarlos.