

## Ejercicio práctico para preparar el módulo M0491\_3.

ejercicio paso a paso para crear un formulario básico de control de datos de un artículo, utilizando **HTML5**, **CSS** y **JavaScript**.

### Formulario de Datos de Artículo

Vamos a crear un formulario sencillo para capturar información sobre un artículo. Tendrá cinco campos: Nombre del Artículo, Descripción, Precio, Cantidad y Categoría.

#### Paso 1: Estructura HTML (**ForArti.html**)

Primero, crearemos el archivo HTML que contendrá la estructura de nuestro formulario.

#### Algoritmo del Html.

- `<!DOCTYPE html>`: Declara el tipo de documento como HTML5.
- `<html lang="es">`: Define el idioma de la página como español.
- `<head>`: Contiene metadatos.
- `<meta charset="UTF-8">`: Especifica la codificación de caracteres.
- `<meta name="viewport">`: Configura la vista para dispositivos móviles.
- `<title>`: Título que aparece en la pestaña del navegador.
- `<link rel="stylesheet" href="style.css">`: Vincula nuestro archivo CSS para los estilos.
- `<body>`: Contiene el contenido visible de la página.
- `<div class="container">`: Un contenedor para centrar y dar espacio al formulario.
- `<h1>Datos del Artículo</h1>`: Un título para el formulario.
- `<form id="articuloForm">`: El elemento principal del formulario con un id para JavaScript.
- `<div class="form-group">`: Un div para agrupar etiqueta y campo de entrada, facilitando el estilo.
- `<label for="...">`: Asocia un texto descriptivo con un campo de entrada. El atributo `for` debe coincidir con el id del campo.
- `<input type="text">`: Campo de texto para el nombre.
- `<textarea>`: Área de texto para la descripción. `rows="4"` define la altura inicial.
- `<input type="number">`: Campos numéricos para precio y cantidad.
- `step="0.01"`: Permite decimales para el precio.
- `min="0"` y `max="1"`: Establecen valores mínimos.
- `<select>`: Un menú desplegable para la categoría.
- `<option value="">`: Opción inicial que no tiene valor y sirve como placeholder.
- `required`: Atributo HTML5 que hace que el campo sea obligatorio antes de enviar el formulario.
- `<button type="submit">`: Botón para enviar el formulario.
- `<div id="mensaje" class="mensaje">`: Un div para mostrar mensajes al usuario.
- `<script src="script.js"></script>`: Vincula nuestro archivo JavaScript. Se coloca al final del `<body>` para asegurar que el HTML esté cargado antes de que el script intente interactuar con él.

#### Paso 2: Estilos CSS (**ForArti.css**)

Ahora, vamos a darle un poco de estilo a nuestro formulario para que se vea mejor.

#### Algoritmo del CSS:

- `body`: Establece la fuente, color de fondo y centra el contenido en la página usando Flexbox. `min-height: 100vh` asegura que ocupe al menos toda la altura de la ventana.
- `.container`: Define el estilo del contenedor del formulario (fondo blanco, padding, bordes redondeados, sombra y ancho máximo).
- `h1`: Centra el título y le da un color.
- `.form-group`: Añade un margen inferior para separar los campos.
- `label`: Hace que las etiquetas se muestren en una nueva línea (`display: block`), añade un margen inferior, color y negrita.

- input, textarea, select: Estilos generales para los campos de entrada (ancho, padding, bordes, etc.). box-sizing: border-box es crucial para que el padding y el borde no aumenten el ancho total de los elementos.
- textarea: Permite que el usuario redimensione solo verticalmente.
- button[type="submit"]: Estilos para el botón de enviar (fondo azul, texto blanco, padding, etc.) y un efecto :hover para interacción.
- .mensaje: Estilos para el div donde mostraremos los mensajes. display: none lo oculta inicialmente.
- .mensaje.exito y .mensaje.error: Clases para dar estilos visuales diferentes a los mensajes de éxito y error.

### Paso 3: Lógica JavaScript (**ForArti.js**)

Finalmente, vamos a añadir la lógica con JavaScript para manejar el envío del formulario, validar los datos y mostrar un mensaje.

#### Algoritmo del JavaScript:

- document.addEventListener('DOMContentLoaded', () => { ... }); Asegura que el script se ejecute solo cuando todo el HTML ha sido cargado y parseado.
- const articuloForm = document.getElementById('articuloForm');: Obtiene una referencia al formulario usando su id.
- const mensajeDiv = document.getElementById('mensaje');: Obtiene una referencia al div donde mostraremos los mensajes.
- articuloForm.addEventListener('submit', (event) => { ... }); Añade un "escuchador de eventos" al formulario que se activa cuando se intenta enviar.
  - event.preventDefault(); ¡Muy importante! Detiene el comportamiento por defecto del navegador de recargar la página al enviar un formulario. Esto nos permite manejar el envío con JavaScript.
  - **Limpieza de mensajes:** Antes de cada validación, limpiamos cualquier mensaje anterior y reseteamos sus clases.
  - **Obtener valores:** Obtenemos los valores de cada campo del formulario.
    - .value.trim(): Elimina espacios en blanco al principio y al final de los textos.
    - parseFloat() y parseInt(): Convierten los valores de precio y cantidad a números.
  - **Validaciones básicas:**
    - Se crea un array errores para almacenar los mensajes de validación.
    - Se comprueba si los campos obligatorios están vacíos o si los números son válidos/positivos. isNaN() es fundamental para verificar si un valor no es un número.
  - **Mostrar mensajes:**
    - Si hay errores, se añade la clase error al mensajeDiv, se crea una lista <ul> con los errores y se hace visible el div (display = 'block').
    - Si no hay errores, se añade la clase exito, se muestra un mensaje de éxito y se hace visible el div.
    - console.log('Datos del Artículo:', datosArticulo);: Muestra los datos del artículo en la consola del navegador (puedes abrirla con F12 en la mayoría de los navegadores). En una aplicación real, aquí es donde enviarías los datos a un servidor.
    - articuloForm.reset();: Vacía todos los campos del formulario después de un envío exitoso.