



DESARROLLO DE SOFTWARE 2025

FUNDAMENTOS FRONTEND

HTML5 + CSS3 + JAVASCRIPT ES6+

Clase #08 - 26 de Mayo del 2025

Matías Corti

TEMAS

- HTML5 Semántico
 - CSS3 Moderno
 - JavaScript ES6+
 - Demostración integral
-

¿QUÉ DIFERENCIA UNA
PÁGINA WEB DE UNA
APLICACIÓN WEB?



PÁGINA WEB

- Información estática
- Solo lectura
- Navegación simple



APLICACIÓN WEB

- Interactiva
 - Maneja estado
 - Funcionalidades
Complejas
-



HTML



HTML + CSS



HTML + CSS + JavaScript



HTML5

ESTRUCTURA SEMÁNTICA

Estructura ≠ Presentación

Accesibilidad: Lectores de pantalla

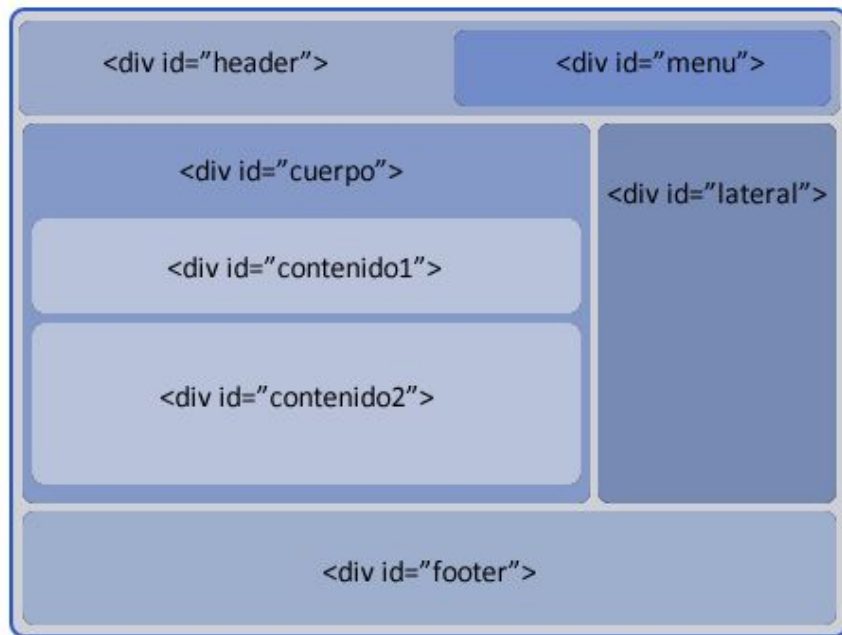
SEO: Mejor indexación

Mantenimiento: Código más claro

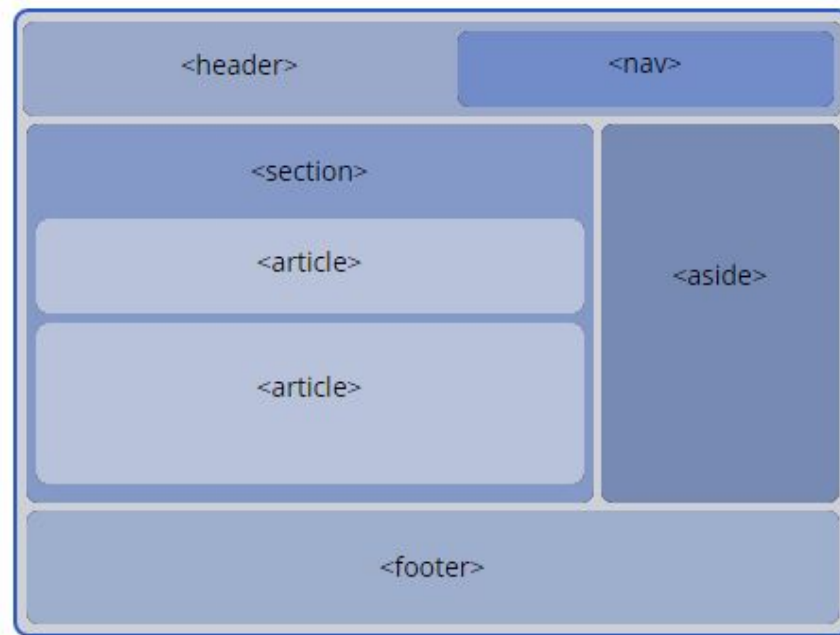
Etiquetas Semánticas

- `<header>` - Cabecera de página/sección
- `<nav>` - Navegación principal
- `<main>` - Contenido principal
- `<section>` - Sección temática
- `<article>` - Contenido independiente
- `<aside>` - Contenido relacionado
- `<footer>` - Pie de página/sección

HTML4



HTML5



HTML 5 - ESTRUCTURA BÁSICA

```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Titulo del Sitio</title>
</head>

<body>
  <header>
    <h1>Titulo Principal</h1>
    <nav><!-- Navegación --></nav>
  </header>
  <main>
    <section><!-- Contenido principal --></section>
  </main>
  <footer><!-- Pie de página --></footer>
</body>

</html>
```




HTML5 FORMULARIOS MODERNOS

Validación sin JavaScript!

Validaciones Nativas y Placeholders

```
<!-- Campo de texto obligatorio con placeholder -->
<input type="text" placeholder="Ingrese su nombre" required>

<!-- Campo de correo electrónico con validación de formato -->
<input type="email" placeholder="usuario@ejemplo.com"
required>

<!-- Campo de contraseña con longitud mínima -->
<input type="password" placeholder="Mínimo 6 caracteres"
minlength="6" required>

<!-- Campo numérico con valores mínimo y máximo -->
<input type="number" placeholder="Edad (18-99)" min="18"
max="99" required>

<!-- Campo de fecha con rango de fechas permitido -->
<input type="date" min="2000-01-01" max="2025-12-31" required>

<!-- Campo de archivo que solo acepta imágenes -->
<input type="file" accept="image/*" required>
```



CSS3 DISEÑO MODERNO

De esto...

Página de 1990

...a esto

Aplicación Moderna 2025

Diseño limpio, responsive, interactivo

Conceptos Importantes

Box Model:

margin, border, padding, content

Flexbox:

Layout flexible y responsive

CSS Grid:

Layouts complejos en 2D

Variables CSS:

Reutilización y mantenimiento

Media Queries:

Responsive design



CSS3

VARIABLES

(CUSTOM PROPERTIES)

- ✓ Fácil mantenimiento
- ✓ Consistencia
- ✓ Temas dinámicos

Definición

```
:root {  
  --primary-color: #3498db;  
  --secondary-color: #2c3e50;  
  --border-radius: 8px;  
  --shadow: 0 2px 10px rgba(0, 0, 0,  
0.1);  
}
```

Uso

```
.button {  
  background: var(--primary-color);  
  border-radius: var(--border-radius);  
  box-shadow: var(--shadow);  
}  
  
.header {  
  background: var(--secondary-color);  
}
```



CSS3 FLEXBOX

Adiós floats y positioning!

[MDN Flexbox](#)
[Juego: Flexbox Froggy](#)

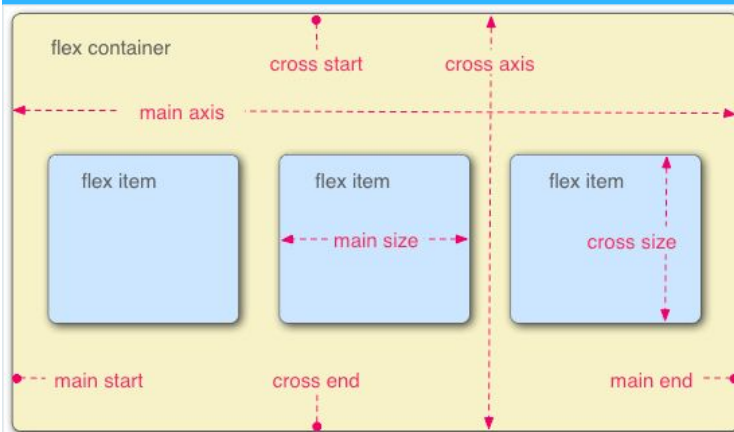
Ventajas

Alineación fácil: vertical y horizontal

Distribución: espacio automático

Responsive: se adapta automáticamente

Orden flexible: sin modificar HTML



```
.container {  
  display: flex;  
  flex-direction: row;  
  /* column */  
  justify-content: center;  
  align-items: center;  
  gap: 1rem;  
}  
  
.item {  
  flex: 1;  
  /* Crece proporcionalmente */  
}
```



CSS3 DISEÑO RESPONSIVE

Un sólo código base



Múltiples dispositivos

[MDN Media Queries](#)

```
/* Mobile First Approach */  
.container {  
  padding: 1rem;  
  flex-direction: column;  
}  
  
/* Tablet */  
@media (min-width: 768px) {  
  .container {  
    padding: 2rem;  
    flex-direction: row;  
  }  
}  
  
/* Desktop */  
@media (min-width: 1024px) {  
  .container {  
    max-width: 1200px;  
    margin: 0 auto;  
  }  
}
```



DEMO: TASKMANAGER

HTML5 + CSS3



JAVASCRIPT

ES6+

SINTAXIS MODERNA

[MDN JavaScript](#)

Variables

```
// ❌ Evitar  
var nombre = 'Juan';  
  
// ✅ Usar  
const nombre = 'Juan'; // Inmutable  
let edad = 25; // Mutable
```

Template Literals

```
// ❌ Concatenación  
const mensaje = 'Hola ' + nombre +  
' , tienes ' + edad + ' años';  
  
// ✅ Template literals  
const mensaje = `Hola ${nombre},  
tienes ${edad} años`;
```

[MDN Template Literals](#)

Arrow Functions

```
// ❌ Función tradicional  
function suma(a, b) { return a + b; }  
  
// ✅ Arrow function  
const suma = (a, b) => a + b;
```

[MDN Arrow Functions](#)

Destructuring

```
// ✅ Extraer propiedades  
const { name, email } = usuario;  
const [primero, segundo] = array;
```

[MDN Destructuring](#)




JAVASCRIPT

ES6+

DOM Y EVENTOS

[MDN DOM](#)

Selección de Elementos

```
//  Métodos modernos
const elemento = document.querySelector('#mi-boton');
const elementos = document.querySelectorAll('.card');

// Event Listeners
elemento.addEventListener('click', (e) => {
  console.log('Botón clickeado'); });
```

Manipulación del DOM

```
// Crear elementos
const div = document.createElement('div');
div.className = 'task-item';
div.innerHTML = ` <h3>${titulo}</h3> <p>${descripcion}</p>
`;

// Agregar al DOM
container.appendChild(div);
```




JAVASCRIPT

ES6+

FETCH API

CONSUMO DE DATOS

[MDN Fetch](#)

```
// Función async/await
const cargarDatos = async () => {
  try {
    const response = await
    fetch('https://jsonplaceholder.typicode.com/users' );

    if (!response.ok) {
      throw new Error(`Error HTTP:
    ${response.status}`);
    }

    const datos = await response.json();

    console.log(datos);

  } catch (error) {
    console.error('Error al cargar datos:', error);
    alert('No se pudieron cargar los datos' );
  }
};
```



DEMO: TASKMANAGER

HTML5 + CSS3 + JS



RECURSOS Y REFERENCIAS



Documentación

- [MDN Web Docs](#)
- [JavaScript.info](#)
- [CSS-Tricks](#)



Herramientas

- VS Code
- Chrome DevTools



Práctica

- [CodePen](#)
 - [Flexbox Froggy](#)
 - [Grid Garden](#)
-

TAREAS

TAREA PARA LA SEMANA

Hasta el 31/05/2025
(inclusive)

Maquetación inicial del proyecto (Grupal)

Repartir con el equipo la participación de todos.

1. Crear en el repositorio una carpeta llamada "Maquetación" para subir los archivos necesarios para la actividad.
2. Incluir estructura HTML completa de la pantalla principal del proyecto
3. Utilizar hojas de estilos CSS modernos: con Flexbox, variables CSS y responsive design con media queries.
4. Incluir un archivo de Scripts JavaScript (ES6+), e implementar al menos 2 interacciones en la maqueta (botones, formularios, etc.)
5. Partiendo de esa pantalla, crear al menos otras dos secciones importantes del sitio, utilizando el mismo layout, y compartiendo archivos hojas de estilo y scripts. Verificando que todo funcione.
6. Avisar al profesor por el canal del Grupo de Slack una vez completado y subido al repositorio de GitHub.