

# Mis notas en clase

---

Durante el primer semestre en la materia de Desarrollo de Software Web, estos son los temas y conceptos que considero importantes y que fui aprendiendo paso a paso. Traté de entender cada punto como si fuera parte de un camino que me va llevando desde lo más básico hasta lo más útil para crear sitios web simples pero funcionales.



## 1. Fundamentos del desarrollo web

- Aprendí que un sitio web es como una casa digital: tiene una dirección (URL), un plano (HTML), decoración (CSS) y funciones internas (JavaScript).
- El **front-end** es lo que vemos (colores, botones, textos) y el **back-end** es lo que pasa detrás (bases de datos, lógica del servidor).
- Cuando escribimos una dirección en el navegador, este envía una solicitud al servidor, que responde con los archivos del sitio web.
- Los **sitios estáticos** son fijos, no cambian (como una foto), mientras que los **dinámicos** pueden mostrar contenido diferente (como Facebook o un blog).



## 2. HTML - La estructura de las páginas

- HTML es el lenguaje que usamos para armar la estructura de una página web. Todo comienza con `<html>`, `<head>` y `<body>`.
- Con etiquetas como `<h1>` hasta `<h6>` se agregan títulos; con `<p>` se escriben párrafos; `<img>` para imágenes y `<a>` para enlaces.
- Aprendimos que usar etiquetas correctas (como `<nav>`, `<section>`, `<footer>`) ayuda a que la web sea más clara y accesible.
- También vimos cómo crear formularios con `<form>`, `<input>`, `<textarea>` y botones, muy útiles para que el usuario pueda interactuar.



## 3. CSS - El diseño visual

- CSS es lo que le da estilo a la web: colores, tamaños, espacios, etc.
- Usamos propiedades como `color`, `margin`, `padding`, `border`, y `font-size` para hacer que la web se vea mejor.
- Las **clases** (`.nombre`) y **IDs** (`#nombre`) ayudan a aplicar estilos específicos a partes del HTML.
- Aprendimos a hacer sitios que se adaptan a diferentes pantallas (celulares, tablets) usando **media queries**.
- Con **Bootstrap** pudimos trabajar más rápido, usando clases ya hechas para botones, tarjetas, columnas y más.

---

## 4. JavaScript - Interactividad básica

- JavaScript permite que la página **reaccione** a lo que hace el usuario.
- Se conecta al HTML con la etiqueta `<script>` o desde archivos externos `.js`.
- Con JavaScript mostramos mensajes (`alert()`), cambiamos texto (`innerHTML`) o estilos (`style.color`).
- Detectamos eventos como clics (`onclick`) o teclas (`onkeydown`) y respondemos con funciones.
- También usamos **condicionales** (`if`, `else`) y **funciones** para hacer cálculos o validaciones básicas.

---

## 5. Organización de un sitio web

- Para que todo funcione bien, hay que ordenar los archivos: una carpeta para imágenes (`img`), otra para estilos (`css`), y otra para scripts (`js`).
- Nombrar los archivos sin espacios ni mayúsculas evita errores: por ejemplo, `pagina.html`, `estilos.css`, `script.js`.
- Es importante mantener el código limpio, con sangrías, saltos de línea y sin repeticiones innecesarias.
- Los **comentarios** (`<!--` en HTML `-->`, `//` en JS) nos ayudan a recordar para qué sirve cada parte del código.

---

## 6. Primeros proyectos personales

- Uno de nuestros trabajos fue crear una **página personal** para presentarnos como estudiantes.
- Usamos **Bootstrap** para organizar el diseño con tarjetas, columnas y estilos bonitos sin tanto código.
- Agregamos nuestras redes sociales, una pequeña biografía, y hasta un acordeón con nuestras habilidades y formación.
- También aprendimos cómo **subir archivos** HTML y ejemplos simples como tareas de clase o trabajos en Java.

Estas son mis notas hasta ahora. Sé que todavía tengo mucho que mejorar, pero con práctica y ganas, cada tema empieza a tener más sentido.