

## HAZLO TÚ MISMO – 1

En primer lugar, se plantea un ejercicio que se puede trabajar en grupo. Se trata de comprender el papel que juegan HTML5, CSS3 y Javascript en el proyecto, además de cómo se integran estos lenguajes.

En este caso, trabajaremos a nivel del diseño conceptual y técnico, sin necesidad de programar. El objetivo no es escribir código en este caso (para esto está la propuesta 2), sino pensar como desarrolladores web: cómo estructurar el contenido, cómo presentarlo visualmente y cómo hacerlo interactivo.

### PASO 1 – ESTRUCTURA DEL SITIO

Diseñad en papel la estructura básica de una página web sencilla para un tema libre: una tienda online, un blog, una galería de fotos o un perfil personal.

Incluid en vuestro diseño:

- Las principales secciones de la página (ej.: *header*, *nav*, *main*, *aside*, *footer*).
- Dentro de cada sección, qué elementos de contenido tendría (ej.: títulos, párrafos, imágenes, enlaces, etc.)
- Pensad qué etiquetas HTML usaríais para representarlos.

Como resultado, se espera un boceto estructural en papel (llamado *wireframe* o diagrama de hilos) con cajas que representen las secciones y etiquetas principales, así como los enlaces y caminos de navegación entre páginas.

### PASO 2 – PRESENTACIÓN VISUAL

El siguiente paso es decidir cómo se presentará la información que habéis diseñado. Para ello, anotad o dibujad en el esquema:

- Colores principales y secundarios.
- Tipografía (moderna, clásica, formal, informal...).
- Distribución del contenido (columnas, secciones apiladas, menú lateral o superior...).
- Estilos especiales para destacar elementos (botones, enlaces, encabezados, etc.).
- Comportamientos visuales en estados interactivos (*:hover*, *:active*...).

Como resultado obtendremos un diseño visual anotado que indique cómo se vería cada sección y qué estilos se aplicarían con CSS.

### PASO 3 – INTERACTIVIDAD Y COMPORTAMIENTO

Es momento de pensar cómo reaccionará la web frente a interacciones del usuario, o cómo cambiará dinámicamente usando JavaScript. No es necesario escribir código, solo describir el comportamiento.

Anotad, al menos, 2 ejemplos de interacción, como por ejemplo:

- Al pulsar un botón, aparece más información en la página.
- Al rellenar un formulario, se valida la entrada.
- Los productos o entradas se cargan dinámicamente desde una fuente externa (API).

Sería interesante que observarais alguna de las webs que frecuentáis, o similares a las que estáis planteando, y estudiéis qué eventos son capturados. También podéis consultar `UI Events` en *Mozilla Dev* para aprender sobre los eventos existentes.

También hay que pensar en qué información se obtendría en formato JSON desde una API externa y cómo se integraría en la web.

## HAZLO TÚ MISMO – 2

Se pretende crear una página web completa que combine estructura (HTML), estilo (CSS), interactividad (JavaScript) y datos externos (JSON vía API). Como resultado, se generará una pequeña aplicación web funcional que resulte de la integración de estos lenguajes.

### PARTE 1: ESTRUCTURA DEL SITIO CON HTML

Se construirá la base de la web en HTML con etiquetas semánticas.

1. Crea un nuevo archivo llamado 'index.html'.
2. Define la estructura básica del documento HTML5.
3. Añade secciones: <header>, <nav>, <main>, <aside>, <footer>.
4. Dentro del <main>, crea secciones con id='inicio', 'about' y 'api'.

Recuerda:

- Debes utilizar etiquetas semánticas siempre que sea posible.
- Utiliza lugares de referencia para la búsqueda de información.

### PARTE 2: DISEÑO CON CSS

Se va a separar contenido y presentación haciendo uso de los estilos.

1. Crea un archivo 'style.css' en una carpeta 'css/' y enlázalo desde el HTML.
2. Define estilos globales como tipografía, colores o márgenes.
3. Personaliza los títulos, párrafos y secciones usando clases e identificadores.
4. Aplica estilos con al menos un par de selectores avanzados, como pseudoclasas, descendientes, atributos, etc.
5. Usa flexbox para organizar el layout.

### PARTE 3: INTERACTIVIDAD CON JAVASCRIPT

En este paso, vamos a añadir lógica e interacción a partir del DOM.

1. Crea un archivo 'script.js' en la carpeta 'js/' y enlázalo desde el HTML.
2. Muestra un mensaje en la consola al cargar la página con `console.log`
3. Cambia dinámicamente el contenido de un elemento.
4. Crea y añade nuevos elementos al DOM desde JavaScript.
5. Captura eventos del usuario y modifica la interfaz en respuesta.

### PARTE 4: CARGA DE DATOS DESDE UNA API

Repasemos como conectar la web con datos externos en formato JSON.

1. Elige una API pública (por ejemplo, `jsonplaceholder.typicode.com`).
2. Usa `fetch()` en tu script para obtener datos JSON.
3. Inserta los datos obtenidos de la API en algún elemento de la sección con `id="api"`.
4. Aplica estilos a los datos mostrados para mejorar la presentación.

## ENTREGA

Antes de subir el ejercicio, debes comprobar que:

- Ofrece interacciones en botones o animaciones.
- El sitio es responsivo.

Prepara una carpeta que contenga la estructura final del proyecto:

- `index.html`
- `css/style.css`
- `js/script.js`

Genera un ZIP con ello y haz una captura (o más) de la web resultante. Adjúntalos a la respuesta de Moodle en la que, además, respondas a:

- ¿Dónde has encontrado más dificultad?
- ¿Qué te hubiera gustado y no has podido hacer?
- ¿Has hecho algo no visto en la asignatura pero que has consultado en las guías de referencia? Explícalo brevemente.