

# UD6 - ESQUEMA VISUAL – Mapa Conceptual Tema 6: Modelo de Objetos del Cliente (BOM, DOM y Cookies)

## 1. Introducción

- **BOM (Browser Object Model)** → modelo del navegador (historial, ventana, pantalla...).
  - **DOM (Document Object Model)** → representación del documento HTML.
  - JS puede manipular **ambos**, haciendo que la web sea interactiva.
- 

## 2. BOM – Browser Object Model

Permite comunicación con el navegador. No está estandarizado oficialmente, pero todos los navegadores modernos lo implementan.

Elementos principales:

---

### 1) Objeto **window**

Representa la **ventana del navegador**.

Todo lo global pertenece a **window**: variables → propiedades; funciones → métodos.

Incluye al DOM ( **window.document** ).

**Utilidades principales:**

- Tamaño de ventana
- Apertura/cierre de pestañas
- Cuadros de diálogo: **alert** , **confirm** , **prompt**

**Métodos típicos:**

- **window.open()**

- `alert()` , `confirm()` , `prompt()`
  - `setTimeout` , `setInterval` , `clearTimeout` , `clearInterval`
- 

## ◆ 2) Objeto **navigator**

Información sobre el navegador del usuario.

**Propiedades importantes:**

- `userAgent`
  - `language`
  - `onLine`
  - `appName`
  - `appVersion`
  - `cookieEnabled`
  - `platform`
  - `plugins`
- 

## ◆ 3) Objeto **screen**

Datos sobre la **pantalla actual** donde está la ventana.

Propiedades relevantes:

- `width` , `height`
  - `availWidth` , `availHeight`
  - `colorDepth`
- 

## ◆ 4) Objeto **location**

Representa la **URL en uso**.

- Principal: `location.href`
  - Obtener URL
  - Redireccionar asignando una nueva dirección

Otras propiedades:

- `protocol`
  - `host` , `hostname`
  - `pathname`
  - `search` (parámetros GET)
  - `hash`
- 

## ◆ 5) Objeto `history`

Gestiona el historial.

Métodos:

- `history.back()`
  - `history.forward()`
  - `history.go(n)`
- 

## ◆ 6) Temporizadores

1. `setTimeout(func, ms)` – ejecuta una vez
  2. `clearTimeout(id)` – cancela timeout
  3. `setInterval(func, ms)` – ejecuta repetidamente
  4. `clearInterval(id)` – detiene intervalo
- 

# ● 3. DOM – Document Object Model

Permite **modificar contenido, estructura y estilos** del documento.

JS puede:

- Cambiar elementos
  - Cambiar atributos
  - Cambiar estilos
  - Añadir/eliminar nodos
  - Responder a eventos
- 

-

## ◆ Árbol DOM

- El DOM es un **árbol de nodos**.
- Nodo raíz: `document`.

**Tipos de Node ( `nodeType` ):**

- 1 → ELEMENT\_NODE
  - 2 → ATTRIBUTE\_NODE
  - 3 → TEXT\_NODE
  - 8 → COMMENT\_NODE
  - 9 → DOCUMENT\_NODE
- (etc.)

Propiedades relacionadas:

- `nodeName`
  - `nodeValue`
- 

## ◆ Seleccionar nodos

### a) Por ID

`document.getElementById()`

### b) Por etiqueta

`document.getElementsByTagName()`

→ devuelve **NodeList**

### c) Por clase

`document.getElementsByClassName()`

### d) Por selectores CSS

`document.querySelectorAll()`

→ devuelve **NodeList**

---

## ◆ Moverse por el DOM

Propiedades:

- `parentNode`
  - `childNodes`
  - `firstChild` , `lastChild`
  - `nextSibling` , `previousSibling`
  - `children` (solo elementos)
- 

## ◆ **Modificar contenido**

- `textContent` → texto puro
  - `innerHTML` → HTML completo
- 

## ◆ **Crear nodos**

1. `createElement`
  2. `createTextNode`
  3. Unirlos
  4. Insertar en el árbol con:
    - `appendChild`
    - `insertBefore`
    - `replaceChild`
- 

## ◆ **Eliminar nodos**

- `removeChild`
- 

## ◆ **Atributos**

- `hasAttribute`
  - `getAttribute`
  - `setAttribute`
  - `removeAttribute`
  - `toggleAttribute`
-

## ◆ Modificar estilos

- `element.style`
  - `getComputedStyle(element)`
  - `className`
  - `classList` (add, remove, toggle, contains, replace)
- 

## ◆ Atributos data-

Acceso mediante `element.dataset` .

---

# ● 4. Cookies

Pequeños ficheros ( $\leq 4\text{KB}$ ) guardados en el navegador.

### ✓ Crear cookie

```
document.cookie = "clave=valor"
```

### ✓ Codificación

- `encodeURIComponent`
- `decodeURIComponent`

### ✓ Duración

- `max-age=segundos`
- `expires=fecha GMT`

### ✓ Ruta

- `path=/ruta`

### ✓ Borrar cookie

`expires` en una fecha pasada.