DAW
Desarrollo de Aplicaciones Web
2º Curso

DWES
Desarrollo Web Entorno Servidor

UD 1. Introducción a los lenguajes de servidor

1.2 Diseño de páginas Web en cliente

IES BALMIS
Dpto Informática
Curso 2025-2026
Versión 1 (05/2025)

UD1.2 – Diseño de páginas Web en cliente

ÍNDICE

1. Estructura de una página web

Como repaso del módulo de primer curso "**Lenguajes de marcas y sistemas de gestión de información**" veremos algunos aspectos necesarios para el diseño de una página web.

Para crear una página web se usa código HTML, CSS y JS (Javascript) que podemos encontrar organizada de forma:

- **embebida**: todo incluido dentro del mismo archivo incluyendo tanto CSS como Javascript dentro del valor de atributos
- **agrupada**: incluyendo el código CSS en la sección **<style>** y el de Javascript en la sección **<script>**
- **desglosada**: creando un archivo para cada código, es decir, un archivo .htm, otro .css y otro .js

Generalmente trabajaremos con la organización **desglosada** para facilitar el **rendimiento** (mediante el uso de caché) y el **mantenimiento** del código.

Se puede ver un sencillo en ejemplo en:

Práctica – 11html-css
https://riconet.es/fp/DWES/php/indice/?ud=1-2

Aunque no es estrictamente obligatorio, al usar HTML5 es conveniente estructurar la página en secciones utilizando la siguiente plantilla, añadiendo todos los **<section>** según el contenido de la página:

Práctica – 12html5
https://riconet.es/fp/DWES/php/indice/?ud=1-2

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>WEB</title>
</head>
<body>
     <nav>      ...</nav>
     <header>   ...</header>
     <section>  ...</section>
     <section>  ...</section>
     <footer>   ...</footer>
</body>
</html>
```

UTF8

Todos los archivos HTML, CSS y JS son archivos de texto.

Los archivos de texto pueden utilizar diferentes codificaciones pero debemos elegir un estándar de codificación al abrir o guardar un archivo en estas situaciones:

- Compartir archivos de texto con personas que trabajan en otros idiomas
- Descarga de archivos de texto a través de Internet
- Compartir archivos de texto con otros sistemas informáticos

Nosotros usaremos **UTF8**, tal y como remienda la W3C:
https://www.w3.org/International/questions/qa-choosing-encodings.es

Editores

Para trabajar con archivos de texto podemos utilizar cualquier editor que pueda utilizar la codificación UTF8 pero recomendamos **Notepad++** o **VSCode**

Notepad++
https://notepad-plus-plus.org/downloads/ - editor funcional, *simple y cómodo* - ***Portable*** *(https://portableapps.com/apps/development/notepadpp_portable)*

VSCode
https://code.visualstudio.com/download - *editor profesional* - *configurable con* ***Plugins*** - ***Portable*** *(https://code.visualstudio.com/docs/editor/portable)*

2. Código externo embebido

El código HTML puede incluir código externo a través de la etiqueta **<iframe>**.

Se pueden consultar diferentes ejemplos con Youtube y Google Maps,

Práctica – 21iframe
https://riconet.es/fp/DWES/php/indice/?ud=1-2

```
<iframe id="videoYoutube"
            width="1000px"
            height="600px"
            src="https://www.youtube.com/embed/i4RE6dBAjH4"
            frameborder="0"
            allowfullscreen></iframe>

<iframe id="mapsGoogle"
            src="https://www.google.com/maps/d/embed?
mid=165Y3WUmLxwIf_YHXL-YqZHBE55k&hl=en_US"
            width="800px"
            height="600px"></iframe>
```

3. Javascript y JQuery

Cuando deseamos realizar acciones sobre nuestro código HTML o atender eventos del usuario en el cliente usaremos **JS** (Javascript).

Prácticas – 31javascript
https://riconet.es/fp/DWES/php/indice/?ud=1-2

Javascript dispone de estructuras de control como cualquier otro lenguaje de programación. De esta forma, podremos utilizar por ejemplo diferentes instrucciones como **if, for, while, ...**

El **DOM** (Modelo de Objetos del Documento) es una representación en forma de árbol de la estructura de un documento HTML, permitiendo a JavaScript acceder y manipular sus elementos, atributos y contenido. Es una interfaz que facilita la interacción con el documento HTML, permitiendo a los desarrolladores modificar la página web dinámicamente.

Una de las formas para acceder al DOM mediante JS es utilizando el identificador, es decir, el atributo id de una etiqueta mediante el método **document.getElementById()**.

Cuando deseamos atender un evento sobre un objeto del HTML para realizar una acción, como el click del ratón, utilizaremos **document.getElementById().addEventListener()**

Para facilitar estas tareas podemos utilizar bibliotecas ya que nos permitirán escribir un código más simple para realizar las mismas acciones.

jQuery es una biblioteca de JavaScript de código abierto que facilita la creación de páginas web interactivas y dinámicas. Simplifica tareas como la manipulación del Document Object Model (DOM), el manejo de eventos, las animaciones CSS y las llamadas AJAX, permitiendo a los desarrolladores trabajar de manera más eficiente y compatible entre diferentes navegadores

A continuación se muestran varios ejemplos escritos con jQuery.

Prácticas – 32jquery
https://riconet.es/fp/DWES/php/indice/?ud=1-2

4. Tecnologías

Como hemos comentado al inicio, los navegadores funcionan con HTML, CSS y JS, pero existen **librerías**, **extensiones (plugins)** y **frameworks** que nos ayudan a crear una aplicación web.

HTML, **CSS** y **JavaScript** son los tres lenguajes fundamentales para el desarrollo web.

- **HTML** define la estructura y el contenido de una página web,
- **CSS** se encarga del diseño y la presentación visual, y
- **JavaScript** añade interactividad y dinamismo.

jQuery es una biblioteca de JavaScript de código abierto que facilita la creación de páginas web interactivas y dinámicas.

Simplifica tareas como la manipulación del Document Object Model (DOM), el manejo de eventos, las animaciones CSS y las llamadas AJAX, permitiendo a los desarrolladores trabajar de manera más eficiente y compatible entre diferentes navegadores

Bootstrap es un framework front-end de código abierto para el desarrollo web, que facilita la creación de sitios web y aplicaciones responsivas.

Proporciona una colección de herramientas y componentes pre-diseñados basados en HTML, CSS y JavaScript para acelerar el proceso de desarrollo y garantizar la consistencia del diseño en diferentes dispositivos

Las **extensiones** y **plugins** son herramientas, utilizadas en desarrollo web con HTML, CSS y JavaScript, que añaden funcionalidades extra a tu entorno de desarrollo, permiten mejorar la productividad, automatizar tareas y facilitar la depuración, entre otras cosas.

Utilizando estas tecnologías básicas del desarrollo de aplicaciones web en cliente, y que serán usadas y explicadas en profundidad en otros módulos, veamos algunos ejemplos.

Prácticas – 41html-bootstrap-local
https://riconet.es/fp/DWES/php/indice/?ud=1-2

En estos ejemplos, se cargan todos los archivos del dominio local, de forma que el desarrollador ha tenido que descargar previamente todo el software utilizado como jQuery, Bootstrap, Aos, Swiper, ...

CDN (Content Delivery Network) o Red de Entrega de Contenido

Para incluir las librerías, extensiones y frameworks podemos usar links externos o locales.

Una **CDN**, o Red de Entrega de Contenido (Content Delivery Network), es un sistema de servidores distribuidos geográficamente que almacenan copias de contenido web (como imágenes, videos, archivos CSS y JavaScript) cerca de los usuarios finales.

Esto permite que el contenido se cargue más rápido y de manera más eficiente, mejorando la experiencia del usuario y reduciendo la carga en el servidor original.

A continuación se muestra el mismo ejemplo pero cargando todo el software de servidores CDN extrernos.

Prácticas – 42html-bootstrap-external
https://riconet.es/fp/DWES/php/indice/?ud=1-2

5. Sensores y hardware del dispositivo

Desde JavaScript en un navegador web, puedes acceder a varios sensores y servicios del dispositivo, pero con limitaciones de seguridad y permisos.

A continuación se muestra un resumen utilizando **navigator**, **window** y **Notification**:

Sensores del Dispositivo

- **Geolocalización**
  - `navigator.geolocation.getCurrentPosition()`
  - Requiere permiso del usuario (`https` en la mayoría de los navegadores).
- **Sensor de Movimiento**
  - Requiere `https` y algunos navegadores (como Safari) piden permiso.
  - **Acelerómetro**
    - `window.addEventListener("devicemotion", ...)`
  - **Giroscopio**
    - `window.addEventListener("deviceorientation", …)`
- **Sensor de Luz Ambiental**
  - `window.addEventListener("devicelight", …)`
- **Sensor de Proximidad**
  - `window.addEventListener("userproximity", …)`
- **Magnetómetro (Brújula)**
  - Parte de `deviceorientation` en algunos navegadores.

Hardware y Servicios del Sistema

- **Cámara y Micrófono**
  - Requiere permiso explícito del usuario.
  - `navigator.mediaDevices.getUserMedia({video:true,audio:true})`
- **Batería**
  - API obsoleta, pero aún funciona en algunos navegadores
  - `navigator.getBattery()`
- **Estado de Red**
  - para ver tipo de conexión: 4G, WiFi, etc.
  - `navigator.connection`
- **Bluetooth**
  - Web Bluetooth API, soporte limitado
  - `navigator.bluetooth.requestDevice()`
- **USB**
  - requiere `https` y permisos
  - `navigator.usb.requestDevice()`
- **NFC**
  - soporte experimental.
  - `navigator.nfc`

- **Vibración**

    - soporte en móviles
    - `navigator.vibrate()`
- **Pantalla (Información)**
    - `screen.width, screen.height, screen.orientation`
- **Clipboard (Portapapeles)**
    - requiere permiso
    - `navigator.clipboard.readText() / writeText()`
- **Notificaciones**
    - envía notificaciones push
    - `Notification.requestPermission()`

A tener en cuenta que algunas de estas tecnologías o APIs:

- requieren https
- no funcionan en localhost
- necesitan permiso explícito del usuario
- sólo funcionan en móviles
- tienen soporte limitado en navegadores (como Bluetooth/USB)

A continuación se muestra un ejemplo obteniendo la ubicación mediante geolocalización.

Prácticas – 51sensores
https://riconet.es/fp/DWES/php/indice/?ud=1-2

6. AJAX

AJAX, que significa Asynchronous JavaScript and XML, es una técnica de desarrollo web que permite a las páginas web actualizarse de forma asíncrona, sin necesidad de recargar la página completa.

AJAX utiliza **JavaScript** y texto en formatos estándar (como **JSON** o **XML**) para comunicarse con un servidor en segundo plano y actualizar solo las partes necesarias de la página.

Para realizar la llamada desde **Javascript** podemos usar **fetch**:

```
fetch(url, {
      method: 'GET', // Método HTTP
      headers: {
       'Content-Type': 'application/json', // Tipo de dato enviado
       'Accept': 'application/json',       // Tipo de dato esperado
      },)
  .then(response => response.json())
  .catch(err => console.log(err))
```

También se puede utilizar **JQuery** con **$.ajax** en vez de Javascript directamente para facilitar el tratamiento de los datos recogidos en la llamada al servidor.

```
$.ajax({
  url: url,
  method: 'GET',
  dataType: 'json',
  success: function(data) {
    console.log(data); // Respuesta exitosa
  },
  error: function(err) {
    console.log(err); // Manejo de errores
  }
});
```

A continuación se muestran varios ejemplos:

ajax_code_ip.html

Práctica – 61ajax
https://riconet.es/fp/DWES/php/indice/?ud=1-2

Se accede mediante AJAX al APIREST de
https://ipinfo.io/json
mostrando:

- el JSON completo recibido
- el campo IP
- el texto del Response Status
- el código del Response Status

ajax_code_libros_status.html

Práctica – 61ajax
https://riconet.es/fp/DWES/php/indice/?ud=1-2

Se accede mediante AJAX al APIREST de
https://riconet.es/fp/apirest/libros/3
mostrando:
- el JSON completo recibido
- el campo titulo
- el texto del Response Status
- el código del Response Status

ajax_libros_apirest_lista.html

Práctica – 61ajax
https://riconet.es/fp/DWES/php/indice/?ud=1-2

Se muestra una tabla vacía con un botón de "**Carga Libros**"
Al pulsar el botón se accede mediante AJAX al APIREST de
https://riconet.es/fp/apirest/libros
cargando en la tabla el JSON de libros recibido.

ajax_libros_apirest.html

Práctica – 61ajax
https://riconet.es/fp/DWES/php/indice/?ud=1-2

Se muestra una tabla vacía con dos botones:
"**Carga Libros**" y "**Nuevo libro**"

Al pulsar el botón "**Carga Libros**" se accede mediante AJAX al APIREST de
https://riconet.es/fp/apirest/libros con GET
cargando en la tabla el JSON de libros recibido.

Al pulsar el botón "**Nuevo Libro**" se accede mediante AJAX al APIREST de
https://riconet.es/fp/apirest/libros con POST
insertando un nuevo libro con datos automáticos numerados.
Después de insertar se deberán recargar los libros para refrescar los datos actuales existentes.

ajax_traduce.html

Práctica – 61ajax
https://riconet.es/fp/DWES/php/indice/?ud=1-2

Este ejecmplo permite traducir textos del español al inglés y viceversa usando el APIREST:
https://api.mymemory.translated.net

7. Diseño avanzado

Para realizar un diseño avanzado necesitaremos tener en cuenta aspectos que proporcionarán características profesionales a nuestro resultado final.

7.1 Diseño Responsive, Animaciones y Control de Eventos

Responsive (adaptable o responsivo) se refiere a la capacidad de un sitio web para ajustarse y mostrarse correctamente en diferentes dispositivos y tamaños de pantalla, como ordenadores, tablets y smartphones.

Un diseño **responsive** asegura que el contenido y la estructura de la página se adapten automáticamente al tamaño y orientación de la pantalla del dispositivo (móvil, tablet, escritorio).

Se basa en:

- **Media Queries**: Reglas CSS que aplican estilos según el ancho de pantalla.
- **Layouts Flexibles**: Uso de unidades relativas (%, vw, rem) en lugar de fijas (px).
- **Imágenes Adaptables**: Técnicas como srcset para cargar imágenes optimizadas según el dispositivo

Ejemplo
para adaptar el tamaño de la fuente en diferentes tamaños de pantalla

```
/* Mobile (default) */
body { font-size: 16px; }

/* Tablet (768px+) */
@media (min-width: 768px) {
  body { font-size: 18px; }
}

/* Desktop (1024px+) */
@media (min-width: 1024px) {
  body { font-size: 20px; }
}
```

Las animaciones consisten realizar movimientos de los objetos o cambios en sus características utilizando transiciones, lo que mejora la presentación e interactividad. Pueden aplicarse a diferentes eventos como:

- al cargar la página,
- al aparecer haciendo scroll,
- al hacer click
- ...

Se deben usar con moderación para no afectar rendimiento.

Los eventos son acciones del usuario (click, scroll, teclado) o del navegador (carga de página) y pueden controlarse mediante Javascript para crear interacciones dinámicas sin recargar la página.

Algunos de los eventos que suelen controlarse son:

- click
- scroll
- resize
- keydown

7.2 Optimización de imágenes

Las imágenes no optimizadas ralentizan la carga.
Para un menor tiempo de carga se suelen tener en cuenta:

- **Formato**: usar formatos vectoriales (como SVG) o modernos con mejor compresión (como WebP)
- **Compresión**: reducir su tamaño pero sin perder calidad visible
- **Lazy Loading**: realizar la carga de imágenes solo cuando son necesarias (por ejemplo al hacer scroll).

7.3 URLs amigables

Una URL legible y descriptiva mejoran el SEO y la usabilidad, permitiendo a los usuarios y buscadores entender el contenido antes de entrar.

Consiste en cambiar la estructura de parámetros QUERY por palabras relevantes.
Por ejemplo:

- Web NO amigable: https://dominio/miweb?secc=3&id=123
- Web amigable: https://dominio/miweb/tienda/productos

7.4 Estructura semántica del HTML

HTML5 dispone de etiquetas equivalentes a ** ** de tipo **semántico** que describen el contenido.

Algunas etiquetas semánticas de HTML5 son:

- **<header>**: Cabecera (logo, menú).
- **<nav>**: Navegación principal.
- **<main>/<section>/<article>**: Organizan contenido jerárquicamente.
- **<footer>**: Información en pie de página (contacto, copyright).

El uso de estas etiquetas **mejoran el SEO** porque los buscadores interpretan su estructura y favorecen la accesibilidad ya que los lectores de pantalla navegan eficientemente.

7.5 Validación W3C

El **W3C (World Wide Web Consortium)** es la organización internacional que desarrolla y mantiene los estándares web para garantizar que Internet funcione de manera consistente, accesible y eficiente para todos.

El W3C define estándares web asegurando:

- **Compatibilidad**: funciona en todos los navegadores
- **Calidad**: sin errores de sintaxis.

El W3C proporciona herramientas de validación para revisar nuestro código:

W3C – Herramienta de validación
HTML - https://validator.w3.org/ *Otras herramientas - https://www.w3.org/developers/tools/*

7.6 SEO básico

El SEO (Search Engine Optimization) mejora la visibilidad en buscadores.

El SEO se basa principalmente en:

- **Meta Tags**: **<title>** y **<description>** claros y descriptivos.
- **Jerarquía en etiquetas**: Uso correcto de <h1> a <h6>.
- **Enlaces Internos**: con texto descriptivo (anchor text).
- **URL amigables**: con palabras relevantes del contenido de la web

7.7 Accesibilidad

La accesibilidad web (Web Accessibility) es la práctica de diseñar y desarrollar sitios y aplicaciones que todas las personas puedan usar, incluyendo aquellas con discapacidades visuales, auditivas, motoras o cognitivas. Su objetivo es eliminar barreras tecnológicas para garantizar igualdad de acceso a la información.

Las **Web Content Accessibility Guidelines (WCAG)**, creadas por el W3C, definen 3 niveles de conformidad:

Nivel	Descripción	Ejemplo de Requisito
A (Mínimo)	Accesibilidad básica. Sitio usable con ayudas técnicas.	-Texto alternativo (alt) en imágenes.
AA (Recomendado)	Accesibilidad óptima para la mayoría de casos. Obligatorio en normativas.	-Contraste mínimo de 4.5:1 (texto normal).
AAA (Alto)	Accesibilidad avanzada para entornos especializados.	-Contraste de 7:1 y lenguaje claro (nivel de lectura bajo).

En resumen, para diseñar una web accesible a nivel AA es necesario como mínimo:

- Disponer de un **contraste adecuado de colores**
- Que los **textos sean legibles** (no en imágenes) y tengan una **estructura clara**
- Usar **alt en imágenes**

Aunque existen varias herramientas para comprobar la accesibilidad de un sitio web, para los desarrolladores es muy recomendable usar un plugin en el navegador.

ARC Toolkit es una herramienta de evaluación de accesibilidad web desarrollada por The Paciello Group (TPG), una empresa líder en consultoría de accesibilidad digital. Está diseñada como una extensión para Google Chrome y Microsoft Edge que ayuda a desarrolladores, diseñadores y especialistas en accesibilidad a identificar y corregir problemas de accesibilidad en sitios web y aplicaciones, siguiendo los estándares WCAG (Web Content Accessibility Guidelines).

ARC Toolkit – Herramienta de evaluación
https://www.tpgi.com/arc-platform/arc-toolkit/