

TEMA 2 PARTE 1

LOS LENGUAJES DE LA WEB. HTML Y XHTML

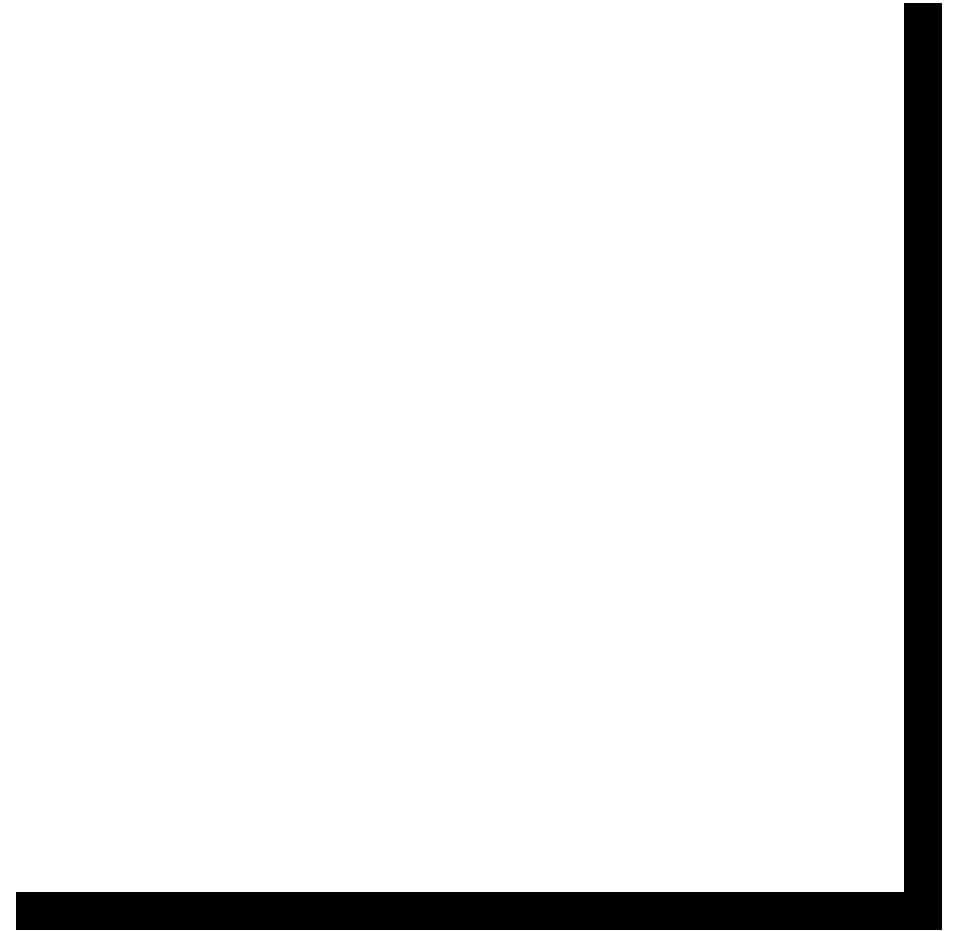
Lenguaje de Marcas y sistemas de Gestión de la
Información

CONTENIDOS

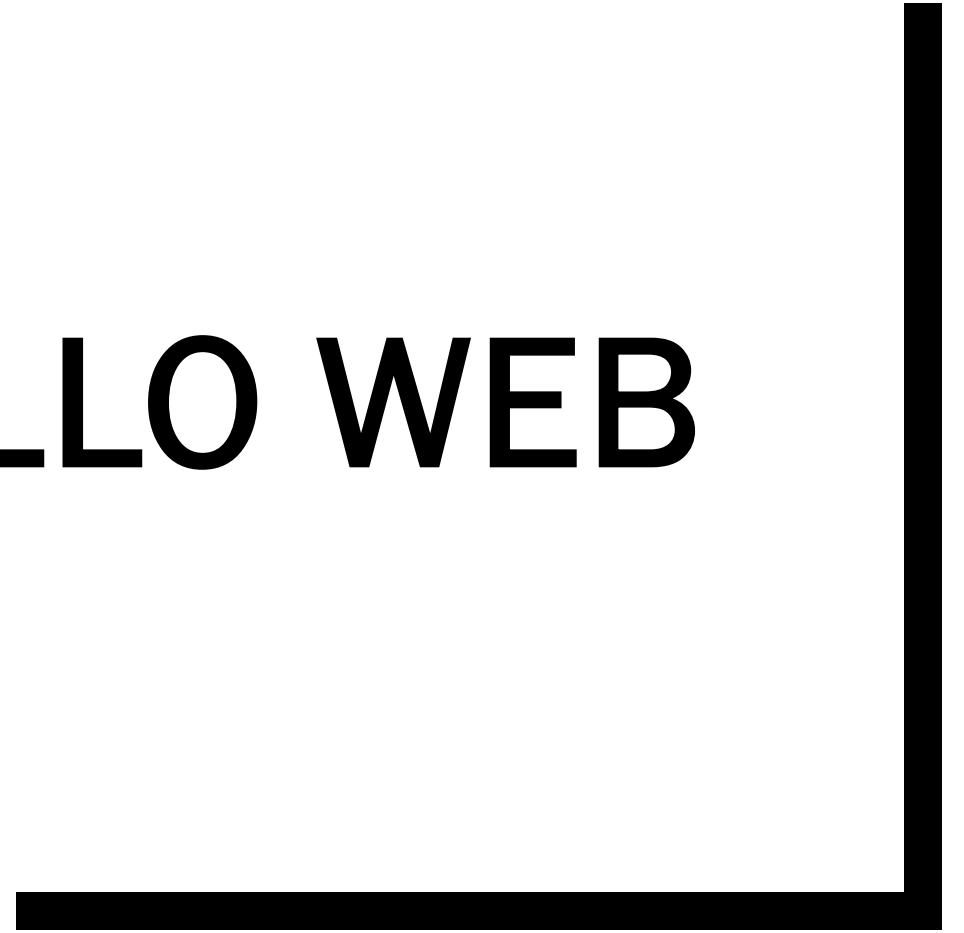
2.1 DESARROLLO WEB

2.2 HTML 5

2.3 HTML 5 - ETIQUETAS DE CABECERA



2.1 DESARROLLO WEB



DESARROLLO WEB

Un desarrollador es un especialista en tecnología de la información (TI) capaz de diseñar, crear e implementar sistemas, programas o aplicaciones informáticas que satisfacen unos objetivos.

Un desarrollador de software o desarrollador web abarca diferentes habilidades para poder cumplir su cometido.

Tradicionalmente, en el terreno de la arquitectura de software y la programación se ha diferenciado de manera general entre dos perfiles de desarrolladores: **Frontend** y **Backend**, las dos caras de la programación

PERFILES DE DESARROLLADORES

FRONTEND

Aquella parte de la programación que está a la vista del usuario, que se ejecuta en la parte del cliente, es decir, la capa de presentación o capa UI (User Interface).

La lógica de la presentación y el diseño de la interfaz de la aplicación son tarea del desarrollador **Frontend**, que también se encarga de la manera en la que el usuario se relaciona con la aplicación a través del navegador.

BACKEND

Alude a aquella parte de la programación que no es visible para el usuario.

Esta parte no se ejecuta en el navegador o dispositivo del cliente, sino en un servidor (físico o en la nube).

El **Backend** se encarga de procesar toda la información, de relacionar los datos en el interior de la aplicación con las capas visuales que ha creado el **Frontend**.

DESARROLLO FRONTEND

Los lenguajes básicos con los que trabaja un desarrollo **Frontend** son **HTML**, **CSS** y **Javascript**. A partir de aquí, se multiplican los frameworks y las herramientas, que se tornan cada vez más complejas.

Cuando se trata de desarrollar un sitio web lo primero que debes considerar es cómo lo quieres construir, como un sitio **web estático** o como un sitio **web dinámico**. Pero ¿qué hace que un sitio web sea categorizado “**estático**” o “**dinámico**”?

Pero antes de entrar en detalles de cada uno de estos tipos de sitios web, primero debes entender cómo funciona la comunicación en Internet cuando queremos ver una página web.

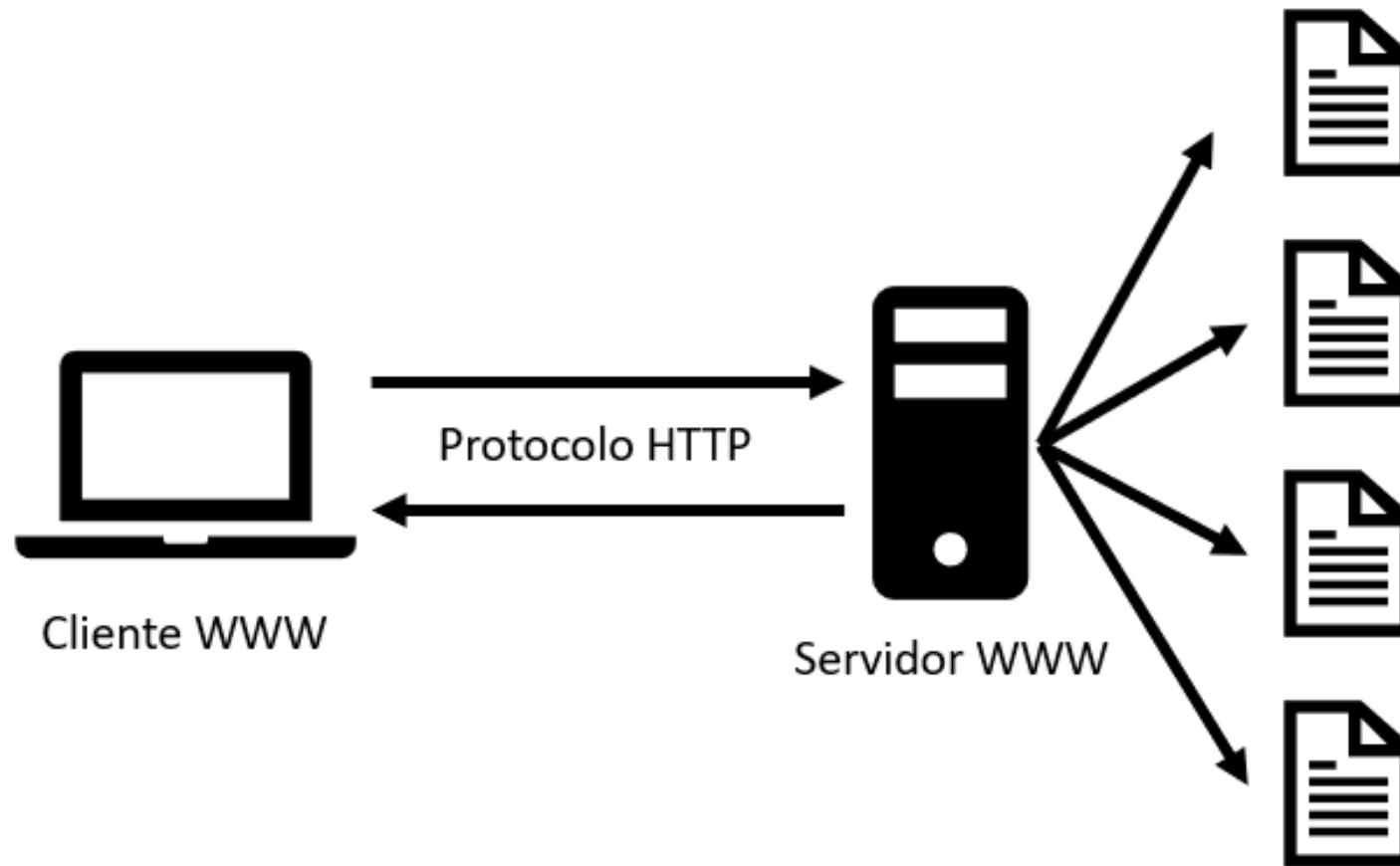
COMUNICACIÓN ENTRE SERVIDORES Y NAVEGADORES WEB

Para entender la comunicación más básica que ocurre en Internet, debemos saber que en ella están involucrados un **servidor web**, como **IIS**, **Apache** o **NGINX**, que contiene los archivos **HTML**, **CSS** y **JavaScript** de las páginas web, y un **cliente**, el navegador (**Chrome**, **Firefox**, **Edge**).

El **servidor web** y el **cliente** se comunican a través de los protocolos **HTTP** (Hypertext Transfer Protocol), el protocolo de Transferencia de Hipertexto y la versión segura **HTTPS** (HyperText Transfer Protocol Secure), una serie de reglas que permiten la transferencia de información a través de archivos en la Internet.

La comunicación entre ellos inicia cuando el usuario, a través del navegador, ingresa la dirección de un sitio web conocida también como la **URL (Uniform Resource Locator)**, generando una petición al servidor web para encontrar los archivos de dicha página en dónde están alojados. Luego, el **servidor web** responde a la petición y devuelve los archivos del sitio web. La respuesta llega al navegador como una copia en formato **HTML** de la página web, y es en este momento cuando el usuario puede verla en su pantalla.

COMUNICACIÓN ENTRE SERVIDORES Y NAVEGADORES WEB



PÁGINA WEB ESTÁTICA

El código fuente del sitio web se encuentra fijo, no se mueve ni cambia de ninguna manera.

Cuando hablamos de “**estático**” también podemos referirnos a que la página web tiene un número fijo de página, es decir, que tal como fue diseñada y almacenada en el servidor web, así mismo la recibe el navegador y la ve el usuario.

Como una página web estática básica está compuesta por elementos como títulos, cuadros de textos, etiquetas, imágenes y otros elementos multimedia, un usuario solo puede interactuar con una página web estática a través de lo que permiten los elementos HTML.

En pocas palabras, en una página web estática, verás la misma información, diseño y contenido cada vez que la visites, a menos que alguien aplique cambios al código fuente de forma manual.

PÁGINA WEB ESTÁTICA

VENTAJAS

- El coste inicial de una página web estática puede ser mucho menor que al de una dinámica.
- Son muy flexibles cuando se trata del diseño.
- Los tiempos de carga son muy rápidos.

DESVENTAJAS

- Una página web estática puede ser más difícil de actualizar.
- Agregar contenido a la página web o realizar actualizaciones puede incurrir en costos adicionales.
- Agregar nuevas páginas o funcionalidades a una web estática puede ser más difícil que hacerlo para una web dinámica.

PÁGINA WEB DINÁMICA

Elementos que cambian continuamente, son interactivos y funcionales, en lugar de ser simplemente informativos. Eso requiere utilizar más que solo código **HTML** y **CSS**.

Los sitios web dinámicos basan su comportamiento y funcionalidad en dos tipos de programación, **Frontend** (del lado del cliente) y **Backend** (del lado del servidor). Las instrucciones del lado del cliente es código **JavaScript** que se ejecuta en el navegador. Mientras que las instrucciones que se ejecutan del lado del **servidor** son instrucciones escritas en lenguajes de scripting o programación, como **ASP.Net**, **PHP**, **Python**, etc. y que son ejecutadas para crear lo que el usuario ha solicitado en su interacción con la página.

Una vez ejecutadas las instrucciones en el servidor, un nuevo **HTTP** response se envía al navegador del usuario para mostrarle lo que ha solicitado. El resultado final es el mismo que en un sitio web estático: una página **HTML** que el usuario ve desde el navegador.

Una página web **dinámica** puede ser **más compleja** cuando hablamos de su diseño y desarrollo, pero también es más versátil cuando se trata de la funcionalidad que ofrece.

PÁGINA WEB DINÁMICA

VENTAJAS

- Puede gestionar información a través de **bases de datos**.
- El contenido se puede gestionar a través de un CMS (Content Management System).
- El coste de mantenimiento es menor.

DESVENTAJAS

- Pueden existir limitaciones en el diseño.
- Puede involucrar un alto coste de construcción iniciales.

2.2 HTML 5



HTML5

HTML (Hypertext Markup Language), es un lenguaje de marcado que sirve para definir la estructura y la semántica de nuestra página web.

HTML fue creado y es mantenido por una organización sin ánimo de lucro llamada **W3C**. El **W3C** es un consorcio formado por más de 400 empresas (entre ellas las que desarrollan los principales navegadores como **Google, Microsoft, Mozilla,...**), etc.

Desde el consorcio trabajan en definir cómo debe evolucionar este lenguaje y otros estándares que conforman la web. Posteriormente los fabricantes de navegadores preparan los mismos intentando conseguir que un código funcione igual en todos los navegadores.

ESTRUCTURA Y DEFINICIÓN DE DOCUMENTO HTML

En primer lugar, se encuentra la definición del tipo de documento, el cual se especifica con la etiqueta `<!DOCTYPE` , seguidamente se encuentra la etiqueta `<html>` que especifica el principio y fin del documento HTML, entre las etiquetas de principio y fin del documento se ubican las etiquetas correspondientes al encabezado y cuerpo del documento cuyas etiquetas correspondientes son `<head>` y `<body>`.

Un ejemplo de código de la estructura básica una página en HTML 5:

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        </head>
```

```
    <body>
```

```
        </body>
```

```
</html>
```

COMENTARIOS

Para poder poner **comentarios** y que no sean interpretados por el navegador, se deben poner de la siguiente forma: <!-- Ejemplo de comentario -->

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <!--<link rel="icon" href="logo.png">-->
```

```
  </head>
```

```
  <body>
```

```
    </body>
```

```
</html>
```


DOCTYPE

Cuando observe el código fuente de cualquier página Web, es normal encontrarse con la etiqueta que sirve para especificar el DTD (Definición del Tipo de Documento) al principio del documento.

Esta línea indica al navegador el tipo de documento que va a leer.

Cada versión especifica las reglas que va a seguir el lenguaje de marcado para que los navegadores muestren el contenido correctamente.

IMPORTANTE → La etiqueta **<!DOCTYPE html>** siempre debe ser la **primera línea** del archivo, sin espacios o líneas que la precedan.

Para realizar la especificación del tipo de documento y que este se corresponda con la definición del lenguaje **HTML 5**, se ha de emplear la siguiente etiqueta.

<!DOCTYPE html>

ETIQUETA <HTML>

Ejemplo con atributo

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

Contenido del documento web, resto de etiquetas...

```
</html>
```

El atributo **lang** en la etiqueta de apertura <html> es el único atributo que necesitamos especificar en **HTML 5**. Este atributo define el idioma humano del contenido del documento que estamos creando, en este caso es por español.

Se utiliza para indicar al navegador que el documento que se visualiza está realizado en el lenguaje **HTML**.

Es la **etiqueta raíz** del resto de etiquetas del lenguaje.

Se ha de tener en cuenta que muchas veces en los documentos que se crean se considera como un buen estilo de diseño, el incluir o especificar el lenguaje natural que se utiliza para crear el documento.

Para realizar esta acción es conveniente añadir el **atributo** que permite especificar el lenguaje, quedando la misma de la siguiente forma.

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

Este atributo sirve para ayudar a los motores de búsqueda y navegadores.

2.2 ETIQUETAS DE CABECERA



ETIQUETA CABECERA <HEAD>

<head>

... elementos o etiquetas que componen el encabezado del documento html....

<title>

<meta>

<link>

<script>

<noscript>

</head>

Una de las partes principales en la estructura de un documento web, es la sección de **cabecera** del mismo, la cual se define mediante la etiqueta <head></head>.

Sirve de contenedor para el resto de las etiquetas que puede contener esta sección.

Entre las etiquetas destacables que puede contener:

- ETIQUETA <TITLE>
- ETIQUETA <META>
- ETIQUETA <BASE>
- ETIQUETA <LINK>
- ETIQUETA <SCRIPT>
- ETIQUETA <NOSCRIPT>

ETIQUETA `<head>` `<TITLE>`

```
<head>  
<title>Título del documento</title>  
</head>
```



Sirve para definir el título del documento, siendo obligatoria en todos los documentos.

El título es el único elemento de la sección encabezado que es visible y normalmente los navegadores lo visualizan en la barra de título de la ventana, o como título de la pestaña.

También es usada por los navegadores para añadir la página a su agenda de **favoritos** con el **título especificado** y por último los motores de búsqueda la utilizan para indexar la página.

El número **máximo de caracteres a mostrar** es de **70**, por lo que, si superamos este número, se sustituirá el texto por **puntos suspensivos**.

ETIQUETA `<head>` `<META>`

```
<head>  
<meta name="keywords"  
content="ejercicios, practicas,  
programación, html, etc." />  
</head>
```

Sirve para definir datos (“metadatos”) sobre el documento web o HTML, estos datos **no son visualizados** por el navegador, y se suelen utilizar para **establecer valores** como palabras claves (“keywords”), **descripción de la página**, autor de la misma, última fecha de modificación, etc.

Los **buscadores** suelen utilizar las **keywords** para establecer **valores de búsqueda** para la **página**, mientras que otros tipos de datos podrían ser utilizados por otros tipos de servicio web.

El formato de la etiqueta es el siguiente:

```
<meta atributo=valor />
```

ETIQUETA <head> **<META>** charset

charset

Este **atributo charset** permite establecer el conjunto de caracteres que ha de utilizar el navegador para visualizar la información del documento **HTML** que se está representando. Los conjuntos de caracteres que más se suelen utilizar son **UTF-8** y **ISO-8859-1**, que se corresponden con la codificación Unicode y el juego de caracteres para el alfabeto **Latín**.

```
<meta charset = "UTF-8 " />
```

ETIQUETA `<head>` `<META>` name, content

name, content

Estos atributos permiten **definir pares de datos**, es decir un nombre de valor con su respectivo valor o contenido.

```
<meta name="application-name" content="Ejemplos" />
```

```
<meta name="author" content="Juan" />
```

```
<meta name="description" content="Ejemplos de HTML" />
```

```
<meta name="generator" content="webstorm" />
```

```
<meta name="keywords" content="ejercicios, practicas, programación,  
html, etc." />
```


ETIQUETA `<head>` `<META>` name, content

| name | Descripción |
|------------------|--|
| application-name | Se utiliza para definir el nombre de la aplicación de la cual forma parte el documento web. |
| author | Se utilizar para definir el nombre del autor del documento web. |
| description | Se utiliza para definir una descripción del documento web, es utilizada por los motores de búsqueda, para establecer resultados. |
| generator | Se utiliza para especificar el nombre del programa con el que se ha creado el documento web. |
| keywords | Se utiliza para indicar una serie de términos o palabras que sirven para establecer categorías en la cual el documento web puede ser incluido. Suele ser utilizada por los motores de búsqueda para establecer búsquedas y categorías de los documentos. |

ETIQUETA `<head>` `<META>` `http-equiv`

Este atributo se suele utilizar para especificar y simular atributos de cabecera del protocolo HTTP. Se utiliza estableciendo un valor para el atributo:

```
<meta http-equiv="refresh" content="400" />
```

```
<meta http-equiv="content-type" content="text/html" />
```

| <code>http-equiv</code> | Descripción |
|----------------------------|---|
| <code>content-type</code> | Se utiliza para especificar el juego de caracteres de codificación del documento. |
| <code>default-style</code> | Se utiliza para especificar la hoja de estilos preferida. |
| <code>refresh</code> | Se utiliza para establecer un intervalo de refresco del documento web, el cual será actualizado por el navegador transcurrido el tiempo especificado. |

ETIQUETA `<head>` `<BASE>`

```
<html>
  <head>
    <base href="http://upv.es/" target="_blank"/>
  </head>
  <body>
    
  </body>
</html>
```

La etiqueta `<base>`, permite especificar una URL por defecto y un destino por defecto para todos los enlaces relativos del documento web.

La etiqueta solo puede aparecer una única vez dentro del documento web y ha de situarse a continuación de la etiqueta de encabezado de documento `<head>`, siendo así la primera etiqueta del encabezado.

Se ha de tener en cuenta también, que es una etiqueta sin cierre, es decir no existe etiqueta de cierre de la misma y por tanto se utiliza el formato abreviado para cerrar la misma.

ETIQUETA `<head>` `<BASE>` href, target

Entre los atributos a destacar que tiene la etiqueta se encuentran los siguientes:

href Este **atributo** permite establecer la URL base para todas las **URL's relativas** (lo veremos **más adelante**) que existen en el documento web. La sintaxis para definir esta etiqueta con el atributo href es la siguiente:

```
<base href="valor URL" />
```

target Este **atributo** permite especificar el destino por defecto para todos los enlaces y formularios del documento web.

```
<base href="valor" target="valor" />
```

| target | Descripción |
|-----------|---|
| _blank | Especifica que el destino sea una nueva ventana del navegador, o una nueva pestaña. |
| _self | Por defecto. Abre el destino en el mismo lugar donde se haya realizado el clic. |
| parent | Abre el destino en un marco asociado. |
| _top | Abre el destino ocupando todo el cuerpo de la ventana. |
| Framename | Abre el destino en el marco del cual se especifica el nombre. |

ETIQUETA `<head>` `<LINK>`

Permite realizar la carga y establecer la relación existente entre el documento web actual y un recurso externo.

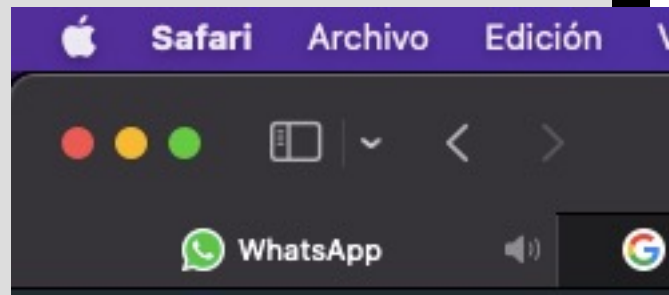
La estructura que nos vamos a encontrar es algo parecido a lo siguiente:

```
<link rel="stylesheet" type="text/css"
href="css/miTema.css" />
```

Esta etiqueta es normalmente muy utilizada para realizar la **carga de hojas de estilo** para el documento web.

La etiqueta se suele insertar en la sección de encabezado del documento web, aunque es posible encontrarla en otros sitios, al final del documento, por ejemplo, lo más adecuado es situarla donde se ha indicado.

También se suele utilizar para poner el **favicon**. El **favicon** es un **pequeño icono** que **identifica** un **sitio web** cuando está abierto en una pestaña o cuando es guardado como favorito.



ETIQUETA `<head>` **<LINK>**

href

Este **atributo** permite establecer la **URL** en la que se encuentra el documento el cual se quiere **enlazar**, con el **documento web**.

```
<link href="valor URL"></link>
```

hreflang

Este **atributo** permite establecer el **lenguaje del documento que se está enlazando**, aunque no realiza ninguna función especial en los navegadores, puede ser **utilizado** por los **motores de búsqueda** para **establecer el idioma**.

```
<link href="URL" hreflang="Codigodelenguaje" ></link>
```

ETIQUETA `<head>` `<LINK>` `rel`

`rel`

El **atributo** sirve para especificar la relación existente entre el documento web y el documento con el que se está realizando el enlace. El atributo es obligatorio y se ha de incluir.

```
<link rel="icon" type="image/x-icon" href="/images/favicon.ico">
```

| rel | Descripción |
|-----------|--|
| alternate | Especifica la relación del documento con una versión alternativa del mismo. (Impresora, Espejo, Traducida, Etc.) |
| author | Especifica la relación del enlace con el autor del documento con el que se enlaza. |
| help | Especifica relación con el documento de ayuda. |

ETIQUETA `<head>` `<LINK>` rel

| rel | Descripción |
|-------------------------|---|
| icon | Se utiliza normalmente para asociar un pequeño icono con el documento web. Este icono suele ser visible en la barra de dirección del navegador. |
| license | Especifica la información de copyright sobre el documento web. |
| next | Especifica que el documento web, es parte de una serie de documentos, e indica que el documento que enlaza es el siguiente en la serie. |
| prefetch
(prerender) | Sirve para especificar desde el documento web que se está cargando, cual es el próximo documento web que se ha de cargar, de esta forma el navegador solicita una petición de cache, cargándose posteriormente la página más rápidamente. |
| prev | Especifica que el documento web, es parte de una serie de documentos, e indica que el documento que enlaza es el anterior en la serie. |
| search | Especifica que el enlace es a la utilidad de búsqueda en el documento. |
| stylesheet | El más utilizado, especifica que el documento con el cual se enlaza es una hoja de estilos a importar, que aplicara formato al documento web. |

ETIQUETA <head> <LINK> type

type

Este **atributo** sirve para especificar el tipo MIME (serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos: texto, audio, vídeo, etc., de forma transparente para el usuario) del documento con el que se esté realizando el enlace.

Se pueden consultar los tipos mime en la siguiente dirección:

<http://www.iana.org/assignments/media-types>

<link type="ValorMime">

Entre los tipos se puede encontrar el de hoja de estilos como es “**text/css**”, o el de texto plano “**text/plain**”, etc.

ETIQUETA `<head>` `<LINK>` media

media

Este **atributo** servirá para indicar el tipo de dispositivo para el cual se está realizando el enlace.

```
<link rel="Valor" type="Valor" href="URL" media="valor">
```

```
<link rel="stylesheet" type="text/css" href="print.css" media="print">
```

| Value | Description |
|--------|---|
| all | Default. Used for all media type devices |
| print | Used for Print preview mode/printed pages |
| screen | Used for computer screens, tablets, smart-phones etc. |
| speech | Used for screenreaders that "reads" the page out loud |

ETIQUETA `<head>` **<SCRIPT>**

Permite la inserción de códigos script, como puede ser **JavaScript** en un documento web del lado del cliente.

Los “navegadores” realizan la carga del documento web de forma secuencial, cuando el navegador se encuentra con una etiqueta de script, puede provocar que si el script es muy lento en su ejecución o si hace referencia a un elemento del documento web, que no ha sido cargado todavía, pueda producirse un error de ejecución en el documento.

Para evitar estas situaciones **es conveniente** situar los scripts al final del documento web, antes de la etiqueta de cierre del mismo.

```
<script>
```

```
document.write('Test prueba')
```

```
document.write('<h1>Hola soy un nuevo documento Web</h1>')
```

```
</script>
```

ETIQUETA `<head>` **<NOSCRIPT>**

Se utiliza para definir un contenido alternativo en el documento web, **cuando el navegador no soporta la ejecución de scripts**, o bien cuando los mismos **están desactivados** en el navegador.

La etiqueta se puede utilizar dentro del encabezado (`<head>`) del documento o bien en el cuerpo del mismo.

Si se utiliza en el encabezado del documento, suele afectar a etiquetas como `<link>`, `<style>` y `<meta>`, de forma que permite realizar la carga alternativa de estos elementos en caso de que el navegador no permite la ejecución de scripts.

Se suele utilizar dentro del cuerpo del documento para indicar que el navegador no permite la ejecución del script que se intenta ejecutar.

`<noscript>`

`Necesitas tener activado Scripts para visualizar correctamente esta página`

`</noscript>`

ETIQUETA `<head>` `<SCRIPT>` y `<NOSCRIPT>`

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
    <h1>The noscript element</h1>
```

```
    <p>A browser with JavaScript disabled will show the text inside the  
    noscript element ("Hello World!" will not be displayed).</p>
```

```
<script>
```

```
    document.write("Hello World!")
```

```
</script>
```

```
<noscript>Sorry, your browser does not support JavaScript!</noscript>
```

```
</body>
```

```
</html>
```

LOS NOMBRES DE CARPETAS Y ARCHIVOS

Con intención de evitar errores es interesante respetar las siguientes normas cuando se asigna nombre a carpetas o archivos:

1. El nombre asignado estará formado por caracteres alfanuméricos: a-z y 0-9
2. No debe contener caracteres no alfanuméricos, signos de puntuación, espacios en blanco, caracteres acentuados, eñes, etc.
3. Los únicos caracteres no alfanuméricos permitidos son el subrayado _ y el guion -
4. No debe superar los 20 caracteres.
5. Utilizar siempre letras minúsculas, aunque se admitan mayúsculas. Hay servidores que distinguen entre mayúsculas y minúsculas. Esto significa que la página Index.htm no es la misma que la página index.htm.
6. Para las páginas **HTML** utilizar siempre la extensión: *.html aunque se admita la *.htm.