

Tecnologías Web

Curso Interfaces de usuario

Teresa Onorati

Mónica Sánchez de Francisco

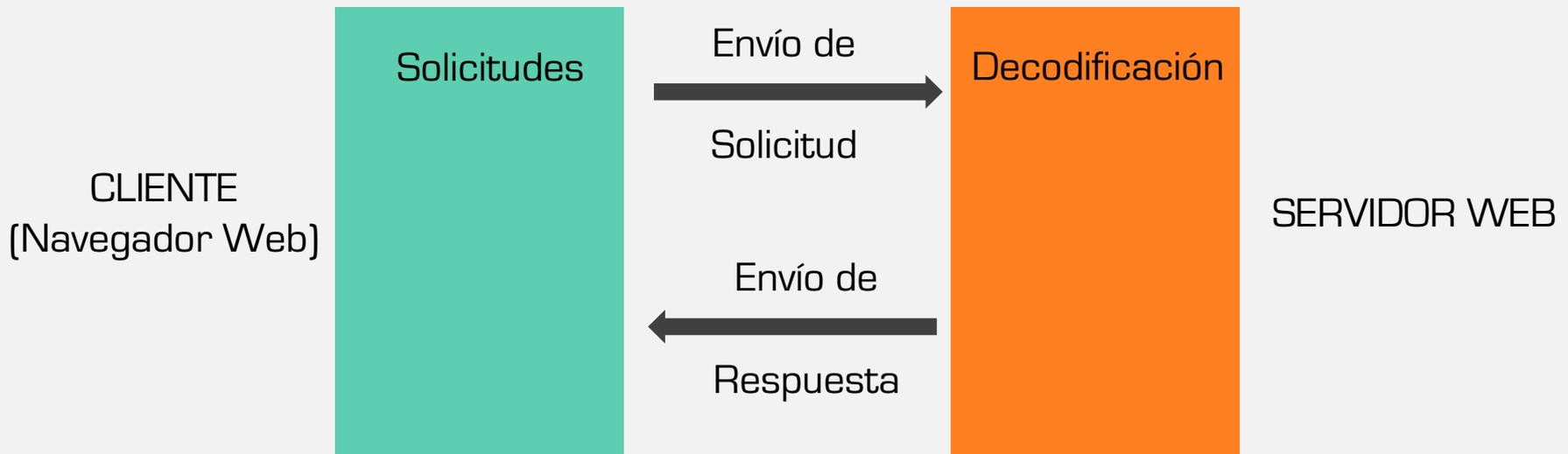
Imagen de [Free Photos](#) en [Pixabay](#)

Internet vs Web

- Internet es una colección de redes de ordenadores interconectadas a escala mundial de millones de computadoras con el conjunto de protocolos propio (TCP/IP).
- La Web es una colección de documentos (hipertextos o hipermedia) interconectados por hipervínculos y URL accesibles a través de Internet.
- Evolución de la web
 - <http://www.evolutionoftheweb.com/>
 - https://upload.wikimedia.org/wikipedia/commons/e/e4/Web_development_timeline.png

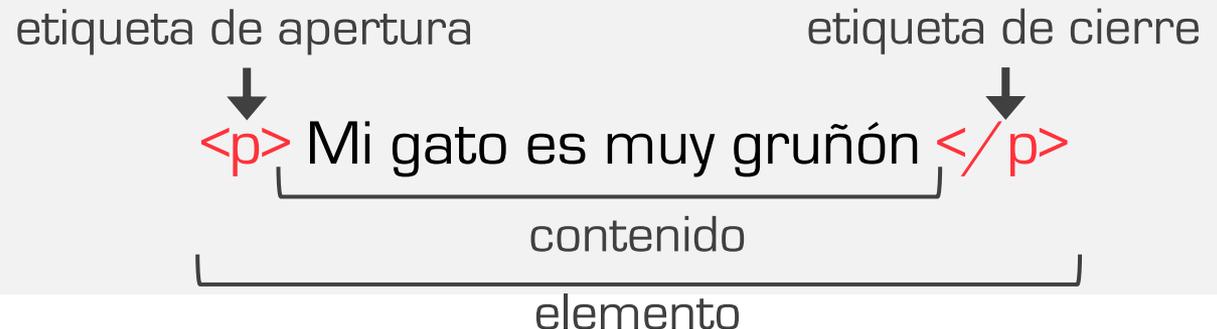
Protocolo HTTP

- El protocolo de transferencia HTTP (HyperText Transfer Protocol) es el sistema mediante el cual se envían las peticiones de acceder a una página web.
- Dispone de una variante cifrada mediante SSL llamada HTTPS.



HTML (HyperText Markup Language)

- Es un lenguaje de Marcado
- Presenta y estructura la información a través de marcas o etiquetas.
 - Ejemplo: `<html>...</html>`
- Las marcas o etiquetas definen la estructura y aspecto del contenido dentro del espacio de visualización
- El W3C desarrolla los estándares necesarios para HTML
- Se define elemento HTML el conjunto de la etiqueta de apertura, la etiqueta de cierre y el contenido.



HTML (HyperText Markup Language)

- Elementos anidados

```
<p> Mi gato es <strong> my bonito </strong> y el tuyo no </p>
```

- Elementos de bloque y elementos de línea

- Los elementos de bloque suelen ser elementos estructurales en la página que representan por ejemplo párrafos, listas, menús de navegación, pies de página, etc.

```
<p> </p>
```

- Los elementos de línea son aquellos que están contenidos dentro de elementos de bloque.

```
<strong> </strong>
```

```
<a href="URL"> </a>
```

- Elementos vacíos: Son una etiqueta única, que sirven generalmente para insertar / incrustar algo en el documento en el lugar donde la incluimos.

HTML (HyperText Markup Language)

- Atributos

```
<p class = "editor-note">
```

Mi gato es más bonito y el tuyo no

```
</p>
```

- Atributos booleanos: pueden tener únicamente un valor, que como norma general es el mismo que el nombre del atributo

```
<input type = "text" disabled = "disabled">
```

HTML (HyperText Markup Language)

- Estructura de un documento HTML

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>My test page</title>
6    </head>
7    <body>
8      <p>This is my page</p>
9    </body>
10 </html>
```

HTML (HyperText Markup Language)

- Caracteres especiales

- Comienza con un signo de ampersand (&) y finaliza con un punto y coma (;)

- Comentario en HTML

`<p> No soy un comentario </p>`

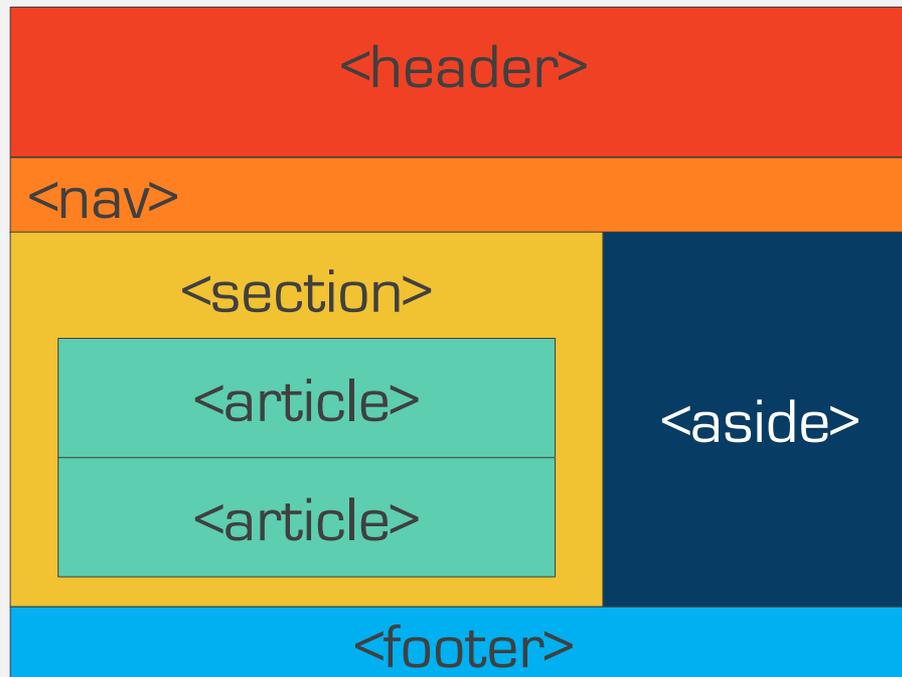
`<!--<p> Sí! soy un comentario </p>-->`

HTML + CSS (Layout)

- CSS (Cascading Style Sheets): se usa para diseñar la forma de presentar los documentos a los usuarios.
- Indica al navegador como se debe visualizar los elementos de un documento HTML.

HTML + CSS (Layout)

- Nuevos elementos HTML 5 (Layout)



CSS (Cascading Style Sheets)

- CSS se puede añadir principalmente de tres formas
 - La primera y la segunda se encuentran en la etiqueta

```
<head>
  <link rel = "stylesheet" href= "css/master.css"/>
  <style type = "text/css"> estilo de la página CSS </style>
</head>
```

- La tercera es con el atributo style directamente en cada

```
<body style="color: red; ">
<p style="font-size: 16px; color: blue; font-family: Arial, Helvetica, sans-serif;"> Mi
primera página </p>
```

CSS (Cascading Style Sheets)

- ESTRUCTURA (Norma de estilo)

H1	{	color	:	green	;	...	}
		propiedad		valor			
		declaración			declaración ; ...		
	BLOQUE de DECLARACIÓN						
NORMA DE ESTILO							

- Comentar en CSS

`/* Esto es un comentario */`

CSS (Cascading Style Sheets)

- Atributo ID: identifica unívocamente un elemento HTML en una página.

HTML → `id="caja"`

CSS → `#caja {color: blue}`

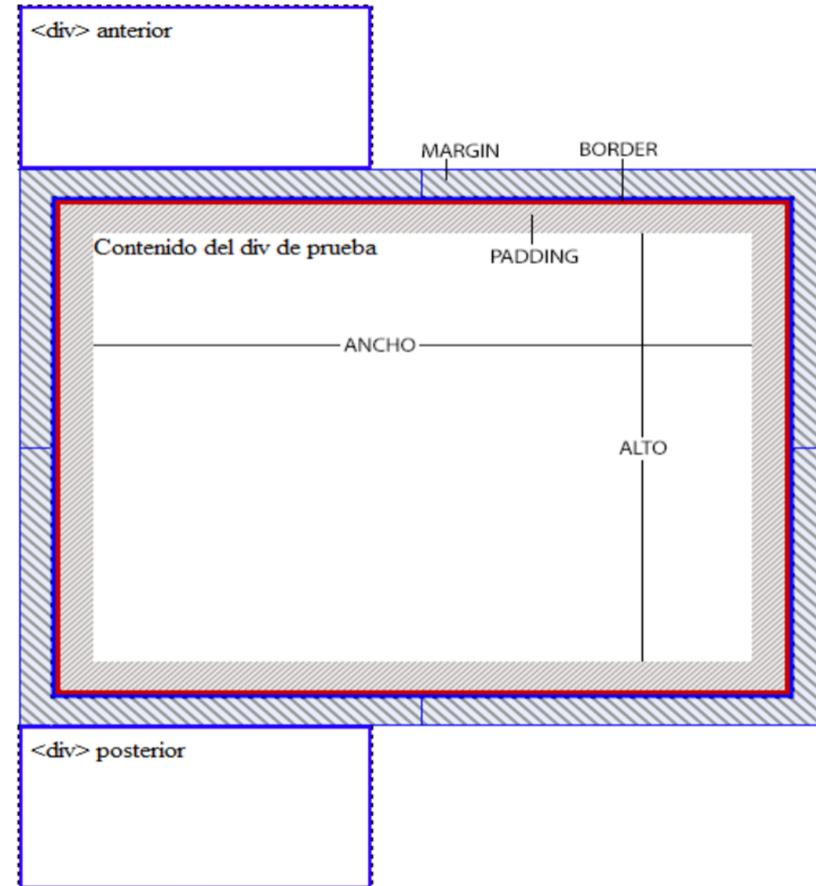
- Atributo CLASS: define una clase de elementos HTML con las mismas propiedades.

HTML → `class="cartas"`

CSS → `.cartas {color: red}`

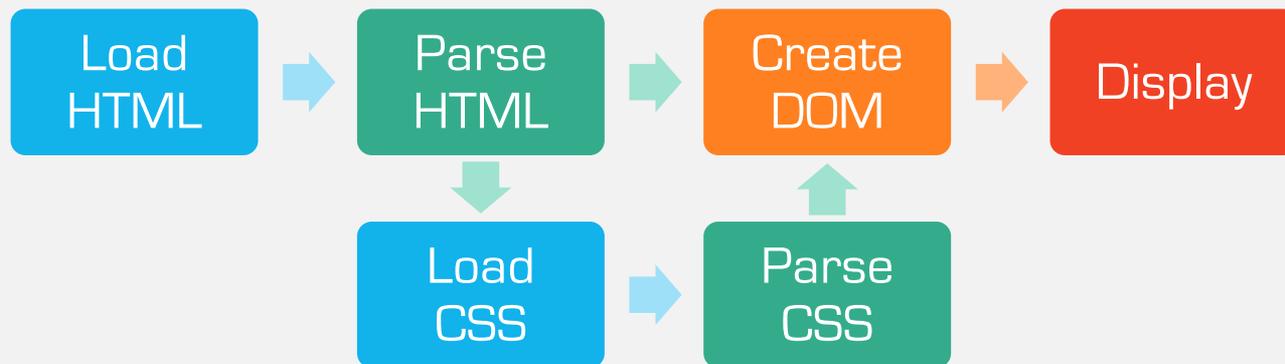
CSS (Cascading Style Sheets)

- Definición de márgenes y bordes de un elemento: Padding, Margin, y Border
- Ejemplo - Etiqueta <div>



CSS (Cascading Style Sheets)

- ¿Cómo funciona CSS?
- Cuando el navegador muestra el documento, debe combinar el contenido y la información de estilos en dos fases
 - El navegador convierte los elementos del HTML y las propiedades de estilo CSS en un DOM (Objeto Documento Modelo)
 - El navegador muestra el contenido del DOM



JS (JavaScript)

- Nace de la necesidad de hacer las páginas dinámicas
- Características
 - Lenguaje de script de cliente
 - Multiplataforma
 - Es un lenguaje de programación interpretado
 - Permite definir la interacción con el usuario
 - Comportamiento dinámico (por ejemplo, control de los campos de formulario)
 - Manipulación de elementos propios del cliente o de la página web

JS (JavaScript)

- Cuando cargas una página Web en tu navegador, tu código (HTML, CSS y JavaScript) es leído dentro de un ambiente de ejecución (pestaña del navegador).
- El navegador lee las líneas de código y lo presenta como el producto final (la página Web).

JS (JavaScript)

- JavaScript funciona con un orden
- Es un lenguaje de interpretación
- ¿Donde colocamos el código JavaScript?
 - En el <body>
`<script> ... </script>`
 - En el <head>
`<script src = MiScript.js> </script>`
 - Evitar ponerlos directamente en el elemento.
- Comentarios:

`// soy un comentario de una línea`

`/* Soy un comentario`

`de multiples líneas*/`

JS (JavaScript)

- Tipos de datos:
 - Números
 - Cadenas, pueden ir con "... " o con comillas simples '...'
 - Booleanos
- Tipos especiales
 - undefined (Indefinido): para indicar que un valor no está definido
 - null: para indicar que no hay ningún valor
 - NaN (Not a Number): indica una indeterminación matemática, por ejemplo, la que resulta de dividir 0 entre 0
 - Infinity (Infinito): indica el valor infinito

JS (JavaScript)

- MÉTODOS
 - `getElementsByTagName()`
 - `getElementById()` // `getElementByClassName()`
 - `querySelectorAll ()`
- Creación de nodos

```
// Crear nodo de tipo Element
var parrafo = document.createElement("p");

// Crear nodo de tipo Text
var contenido = document.createTextNode("Hola Mundo!");

// Añadir el nodo Text como hijo del nodo Element
parrafo.appendChild(contenido);

// Añadir el nodo Element como hijo de la pagina
document.body.appendChild(parrafo);
```

JS (JavaScript)

- FUNCIONES

```
function name (parametro1, parametro2,parametro3){  
  //el código que quedamos ejecutar  
}
```

- EVENTOS

- Relacionados con el ratón
- Relacionados con el teclado
- Relacionados con los formularios
- Relacionados con ventánas y frames
- Relacionados con animaciones y transiciones
- Realcionados con cambios en el DOM
- Relacionados con arrastre de elementos (Drag and Drop)
- Relacionados con audio y video

jQuery

- Es una librería de JavaScript que permite simplificar la manera de interactuar con los documentos HTML
 - modificaciones del DOM, animaciones e interacciones con la técnica AJAX
- Descargar la librería en <https://jquery.com/download>
- Incluirla en el archivo HTML

```
<script type="text/javascript" src="path/to/jquery.min.js"></script>
```

- Antes de interactuar con el DOM hay que asegurarse de que la página se cargue completamente

```
$(document).ready(function(){
```

```
// Aquí va nuestro código para manipular el DOM
```

```
});
```

jQuery

- Función `$()`:

```
$(document).ready(function(){  
    midiv = $('#midiv'); // Devolverá el elemento HTML con id="midiv"  
    mispan = $('.miclase'); // Devolverá el elemento HTML con  
    class="miclase"  
});
```

- Aplicar las funciones que nos ofrece la librería:

```
alert(midiv.text()); // Obtengo el texto de midiv y lo muestro en pantalla  
midiv.html('<h1>' + midiv.text() + '</h1>') // Pongo el contenido de  
// midiv en un tag H1  
miclase.css('color', 'red'); // Le agrego color rojo al texto de miclase
```

jQuery

- jQuery es una librería orientada a plugins:
 - [jQuery UI](#): widgets y efectos visuales para mejorar la usabilidad y la interfaz gráfica de nuestros sitios.
 - [jQuery Form](#): permite serializar formularios y mandar la información al servidor rápidamente.
 - [Fullcalendar](#): permite mostrar y manipular un calendario con diferentes vistas y eventos.
 - [jqGrid](#): permite manipular y crear grillas de forma rápida y sencilla. Además es compatible con dispositivos móviles.
 - [jQuery Mobile](#): similar a jQuery UI, se puede adaptar a dispositivos móviles.
 - [Fancybox](#): permite crear galerías de imágenes usando el atributo rel.
 - ...