

Construcción y diseño de Páginas Web con HTML, CSS y JavaScript

Edición
2023

WWW

Desde www.ra-ma.es podrá
descargar material adicional.

Pablo E. Fernández Casado



Ra-Ma[®]

edü[®]
Ediciones de la U

Construcción y diseño de páginas web con HTML, CSS y JavaScript

Pablo E. Fernández Casado



Ra-Ma®

edü

Conocimiento a su alcance

Fernández Casado, Pablo E.

Construcción y diseño de páginas web con HTML, CSS y JavaScript / Pablo E. Fernández Casado --. Bogotá: Ediciones de la U, 2023

314 p. ; 24 cm

ISBN 978-958-792-585-2 e-ISBN 978-958-792-586-9

1. Informática 2. Programación 3. Introducción a HTML5 4. Introducción al CSS3 5. Introducción a JavaScript 6. I. Tít.
621.39 ed.

Edición original publicada por © Editorial Ra-ma (España)
Edición autorizada a Ediciones de la U para Colombia

Área: Sistemas e informática

Primera edición: Bogotá, Colombia, septiembre de 2023

ISBN. 978-958-792-585-2

© Pablo E. Fernández Casado

© Ra-ma Editorial. Calle Jarama, 3-A (Polígono Industrial Igarza) 28860 Paracuellos de Jarama
www.ra-ma.es y www.ra-ma.com / E-mail: editorial@ra-ma.com
Madrid, España

© Ediciones de la U - Carrera 27 #27-43 - Tel. (+57) 601 6455049
www.edicionesdelau.com - E-mail: editor@edicionesdelau.com
Bogotá, Colombia

Ediciones de la U es una empresa editorial que, con una visión moderna y estratégica de las tecnologías, desarrolla, promueve, distribuye y comercializa contenidos, herramientas de formación, libros técnicos y profesionales, e-books, e-learning o aprendizaje en línea, realizados por autores con amplia experiencia en las diferentes áreas profesionales e investigativas, para brindar a nuestros usuarios soluciones útiles y prácticas que contribuyan al dominio de sus campos de trabajo y a su mejor desempeño en un mundo global, cambiante y cada vez más competitivo.

Coordinación editorial: Adriana Gutiérrez M.

Carátula: Ediciones de la U

Impresión: DGP Editores SAS

Calle 63 #70D-34, Pbx (+57) 601 7217756

Impreso y hecho en Colombia

Printed and made in Colombia

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro y otros medios, sin el permiso previo y por escrito de los titulares del Copyright.

A Pelusa.
Semper fidelis.

ÍNDICE

GLOSARIO DE TÉRMINOS	15
CAPÍTULO 1. CONCEPTOS PREVIOS.....	17
1.1 ENTORNOS DE PROGRAMACIÓN	17
1.2 HISTORIA DE LA WEB	18
1.3 CÓMO FUNCIONA LA WEB	19
1.4 DISEÑO GRÁFICO Y EL PAPEL DEL DISEÑADOR.....	21
1.5 ARQUITECTURA DE LA INFORMACIÓN Y EL PAPEL DEL ARQUITECTO	21
1.6 ESTRUCTURA BÁSICA DE UNA PÁGINA WEB.....	22
1.7 LOS ESTÁNDARES W3C.....	24
CAPÍTULO 2. INTRODUCCIÓN A HTML5.....	25
2.1 INTRUDUCCIÓN	25
2.2 ELEMENTOS BÁSICOS DE HTML	26
2.2.1 Definición del tipo de documento DTD (!DOCTYPE).....	26
2.2.2 Etiqueta html	27
2.2.3 Etiqueta head	28
2.2.4 Etiqueta body.....	28
2.2.5 Comentarios	28
2.3 INFORMACIÓN DEL DOCUMENTO	29
2.3.1 Elemento base	29
2.3.2 Elemento link	29
2.3.3 Elemento meta.....	31
2.3.4 Elemento title	33
2.3.5 Elemento style.....	33
2.3.6 Elemento script.....	33
2.3.7 Principales metadatos a definir.....	34
2.4 ATRIBUTOS GLOBALES.....	36
2.4.1 Atributo accesskey	36
2.4.2 Atributo autocapitalize	36

2.4.3	Atributo autofocus.....	37
2.4.4	Atributo class.....	37
2.4.5	Atributo contenteditable.....	37
2.4.6	Atributo draggable.....	38
2.4.7	Atributo dir.....	38
2.4.8	Atributo enterkeyhint.....	38
2.4.9	Atributo hidden.....	39
2.4.10	Atributo inputmode.....	39
2.4.11	Atributo id.....	40
2.4.12	Atributo lang.....	40
2.4.13	Atributo nonce.....	40
2.4.14	Atributo slot.....	41
2.4.15	Atributo style.....	41
2.4.16	Atributo tabindex.....	41
2.4.17	Atributo title.....	42
2.4.18	Atributo translate.....	42
2.4.19	Atributos personalizados.....	42
2.5	PRACTICA Y JUEGA.....	43
CAPÍTULO 3. DEFINICIÓN DE CONTENIDOS.....		44
3.1	ELEMENTOS SEMÁNTICOS DE HTML.....	45
3.1.1	Elemento article.....	45
3.1.2	Elemento aside.....	45
3.1.3	Elemento div.....	46
3.1.4	Elemento footer.....	46
3.1.5	Elementos h1..h6.....	46
3.1.6	Elemento header.....	47
3.1.7	Elemento hgroup.....	48
3.1.8	Elemento main.....	48
3.1.9	Elemento nav.....	48
3.1.10	Elemento section.....	49
3.2	FORMATEANDO EL TEXTO.....	49
3.2.1	Elemento abbr.....	49
3.2.2	Elemento address.....	50
3.2.3	Elemento bdo.....	50
3.2.4	Elementos blockquote y cite.....	51
3.2.5	Elemento code.....	51
3.2.6	Elemento data.....	51
3.2.7	Elemento dfn.....	52
3.2.8	Elemento em.....	52
3.2.9	Elemento i.....	52
3.2.10	Elementos ins y del.....	52
3.2.11	Elemento kbd.....	53
3.2.12	Elemento mark.....	53
3.2.13	Elemento math.....	53
3.2.14	Elemento pre.....	57
3.2.15	Elementos sub y sup.....	57
3.2.16	Elemento var.....	58
3.3	LISTAS.....	58

3.3.1	Elemento ul	58
3.3.2	Elemento ol	59
3.3.3	Elemento dl	60
3.4	ENLACES.....	61
3.4.1	Elemento a.....	61
3.5	PRACTICA Y JUEGA.....	62
CAPÍTULO 4. INTRODUCCIÓN AL CSS3.....		63
4.1	SOPORTE A LOS NAVEGADORES	63
4.2	/CÓMO FUNCIONA CSS	64
4.3	DEFINICIÓN DE SELECTOR Y CLASIFICACIÓN	64
4.4	UNIDADES DE MEDIDA.....	65
4.4.1	Unidades absolutas o fijas	65
4.4.2	Unidades relativas	66
4.5	SELECTORES.....	67
4.5.1	Simples y combinados.....	68
4.5.2	Pseudo-clases	69
4.5.3	Pseudo-elementos.....	74
4.5.4	Especificidad de los selectores	74
4.6	PROPIEDADES	75
4.6.1	Texto, fuentes y tipos de letra.....	75
4.6.2	Listas	78
4.6.3	Márgenes internos y externos.....	79
4.6.4	Bordes internos y externos	80
4.6.5	Colores	86
4.6.6	Posicionamiento	89
4.6.7	Comportamientos y tamaños.....	92
4.6.8	Reglas arroba y media queries.....	96
4.7	FUNCIONES	104
4.7.1	Funciones de pseudo-elementos.....	105
4.7.2	Funciones de cálculo	105
4.7.3	Funciones gráficas	106
4.8	VARIABLES.....	111
4.9	PRACTICA Y JUEGA.....	113
CAPÍTULO 5. IMÁGENES Y MULTIMEDIA		114
5.1	TIPOS DE IMÁGENES	114
5.2	ELEMENTOS DISPONIBLES EN HTML5.....	115
5.2.1	Elemento audio.....	115
5.2.2	Elementos figure y figcaption.....	115
5.2.3	Elemento img	116
5.2.4	Elemento picture	116
5.2.5	Elemento source	117
5.2.6	Elemento video.....	117
5.3	PROPIEDADES DISPONIBLES EN CSS	118
5.3.1	Propiedad background-attachment	118
5.3.2	Propiedad background-clip	118

5.3.3	Propiedad background-image.....	119
5.3.4	Propiedad background-origin.....	119
5.3.5	Propiedad background-position.....	119
5.3.6	Propiedad background-repeat.....	119
5.3.7	Propiedad background-size.....	120
5.3.8	Propiedad object-position.....	120
5.4	IMÁGENES RECEPTIVAS Y ADAPTATIVAS.....	120
5.4.1	Adaptación de imágenes mediante propiedades CSS.....	121
5.4.2	Adaptación de imágenes mediante consultas de medios.....	123
5.4.3	Selección de imágenes mediante IMG y SRCSET.....	124
5.4.4	Selección de imágenes mediante PICTURE y SOURCE.....	126
5.5	VIDEOS RECEPTIVOS.....	127
5.6	PRACTICA Y JUEGA.....	128
CAPÍTULO 6. DISEÑOS BASADOS EN DIMENSIONES.....		129
6.1	TABLAS.....	129
6.1.1	Elementos disponibles en HTML5.....	129
6.1.2	Elementos disponibles en CSS.....	131
6.1.3	Creación de tablas responsive.....	133
6.2	DISEÑO BASADO EN CAJAS FLEXIBLES (FLEXBOX).....	136
6.2.1	Principales elementos disponibles en CSS.....	136
6.2.2	Creación de flexbox responsive.....	143
6.3	DISEÑO BASADO EN CUADRÍCULAS (GRID LAYOUT).....	144
6.3.1	Principales elementos disponibles en CSS.....	145
6.3.2	Creación de grid responsive.....	147
6.4	PRACTICA Y JUEGA.....	149
CAPÍTULO 7. FORMULARIOS.....		150
7.1	TIPOS DE FORMULARIO.....	151
7.1.1	Formularios de contacto.....	151
7.1.2	Formularios de suscripción.....	153
7.1.3	Formularios de acceso.....	153
7.1.4	Formularios de registro.....	154
7.1.5	Formularios de entrada general.....	155
7.2	ELEMENTOS DISPONIBLES EN HTML5.....	156
7.2.1	Elemento form.....	156
7.2.2	Elemento button.....	158
7.2.3	Elemento datalist.....	159
7.2.4	Elemento fieldset.....	159
7.2.5	Elemento input.....	159
7.2.6	Elemento label.....	163
7.2.7	Elemento legend.....	164
7.2.8	Elemento meter.....	164
7.2.9	Elemento optgroup.....	165
7.2.10	Elemento option.....	165
7.2.11	Elemento output.....	166
7.2.12	Elemento progress.....	166
7.2.13	Elemento select.....	167

7.2.14	Elemento textarea	167
7.3	ELEMENTOS DISPONIBLES EN CSS	168
7.3.1	Propiedad box-sizing	168
7.3.2	Propiedad resize	169
7.4	PRACTICA Y JUEGA	169
CAPÍTULO 8. INTRODUCCIÓN A JAVASCRIPT.....		170
8.1	VARIABLES Y ÁMBITOS	171
8.1.1	Declaración de variables	171
8.1.2	Ámbito de las variables	171
8.2	TIPOS DE DATOS	173
8.2.1	Tipo String	173
8.2.2	Tipo Number	177
8.2.3	Tipo BigInt	185
8.2.4	Tipo Boolean	186
8.2.5	Tipo Symbol	186
8.2.6	Literal null	187
8.2.7	Literales undefined y typeof	187
8.3	OPERADORES Y EXPRESIONES	187
8.3.1	Operadores generales	188
8.3.2	Operadores bit a bit	191
8.4	CONTROL DE FLUJO Y GESTIÓN DE ERRORES	193
8.4.1	Estructura if	193
8.4.2	Estructura if...else	193
8.4.3	Estructura switch	193
8.4.4	Control de errores por tipo de dato	194
8.4.5	Control de errores por presencia	194
8.4.6	Manejo de excepciones	195
8.5	BUCLES Y LA ITERACIÓN	197
8.5.1	Estructura for	197
8.5.2	Estructura for...in	198
8.6	ESTRUCTURA FOR...OF	199
8.7	ESTRUCTURA FOREACH	199
8.8	ESTRUCTURA DO...WHILE	201
8.9	ESTRUCTURA WHILE	202
8.10	SENTENCIA BREAK	202
8.11	SENTENCIA CONTINUE	203
8.12	PRACTICA Y JUEGA	203
CAPÍTULO 9. OBJETOS DE JAVASCRIPT		204
9.1	TIPOS DE OBJETO	204
9.2	PROPIEDADES	205
9.3	MÉTODOS	205
9.4	ARRAYS	206
9.4.1	Creación de arrays	207
9.4.2	Acceso a elementos de un array	207

9.4.3	Inserción y almacenamiento de elementos en un array.....	207
9.4.4	Eliminación de elementos de un array	208
9.4.5	Propiedades	208
9.4.6	Métodos.....	208
9.5	JSON.....	210
9.5.1	Sintaxis.....	211
9.5.2	Creación de JSON.....	211
9.5.3	Acceso a elementos de un JSON.....	212
9.5.4	Inserción y almacenamiento de elementos en un JSON.....	212
9.5.5	Eliminación de elementos de un JSON.....	212
9.5.6	Envío y recepción de JSON.....	212
9.6	ESPECIALES.....	214
9.6.1	El objeto window.....	214
9.6.2	El objeto document.....	215
9.6.3	El objeto Screen.....	217
9.6.4	La interfaz Navigator.....	218
9.6.5	La interfaz Location.....	220
9.6.6	La interfaz HTMLDocument.....	220
9.6.7	El objeto History.....	226
9.6.8	El objeto this.....	229
9.6.9	El objeto globalThis.....	230
9.6.10	El objeto prototype.....	231
9.7	OTRAS COSAS QUE SABER SOBRE LOS OBJETOS DE JAVASCRIPT.....	232
9.7.1	La herencia.....	232
9.7.2	Sentencias get y set.....	234
9.8	PRACTICA Y JUEGA.....	235
CAPÍTULO 10. FUNCIONES EN JAVASCRIPT		236
10.1	CREACIÓN DE FUNCIONES.....	236
10.1.1	Diferencia entre modo estricto o modo no estricto.....	236
10.2	PASO DE PARÁMETROS.....	237
10.2.1	Por asignación directa.....	238
10.2.2	El objeto arguments.....	238
10.3	FUNCIONES ANÓNIMAS.....	239
10.3.1	Ventajas e inconvenientes.....	240
10.4	FUNCIONES CLAUSURA.....	240
10.4.1	Ventajas e inconvenientes.....	241
10.5	FUNCIONES FLECHA.....	242
10.5.1	Ventajas e inconvenientes.....	243
10.6	FUNCIONES ESPECIALES.....	244
10.6.1	Función de prototipo bind.....	244
10.6.2	Función de prototipo call.....	244
10.6.3	Función de prototipo apply.....	245
10.6.4	Diferencias entre call y apply.....	246
10.7	CONTEXTOS Y ENCAPSULAMIENTO.....	247
10.8	PRACTICA Y JUEGA.....	248

CAPÍTULO 11. CLASES EN JAVASCRIPT	249
11.1 CREACIÓN DE CLASES	249
11.2 INSERCIÓN DE MÉTODOS.....	250
11.2.1 Sentencias get y set	251
11.3 EXTENSIÓN DE CLASES.....	252
11.3.1 Extensión a través de especies	253
11.3.2 Extensión a través de super	253
11.4 CLASES ABSTRACTAS Y MIXINS	255
11.5 PRACTICA Y JUEGA.....	255
CAPÍTULO 12. EVENTOS EN JAVASCRIPT.....	256
12.1 PRINCIPIO FUNDAMENTAL DE PROPAGACIÓN	256
12.2 EL OBJETO EVENT	257
12.2.1 Propiedades más frecuentes	257
12.3 LA INTERFAZ TOUCHEVENT.....	260
12.3.1 El objeto Touch.....	260
12.4 LA INTERFAZ KEYBOARDEVENT	261
12.4.1 Propiedades más importantes	262
12.5 LA INTERFAZ MOUSEEVENT	263
12.5.1 Propiedades más importantes	263
12.6 PRINCIPALES MANEJADORES DE EVENTOS.....	264
12.6.1 Eventos de ratón	264
12.6.2 Eventos de formulario	265
12.6.3 Eventos de HTML.....	265
12.6.4 Eventos de tratamiento táctil	266
12.7 OYENTES O LISTENERS	266
12.7.1 Método addEventListener	266
12.7.2 Método removeEventListener	267
12.7.3 Otras formas de establecer listeners	267
12.8 PRINCIPALES EVENTOS DEL DOM	267
12.8.1 Document DOMContentLoaded	267
12.8.2 Window load	268
12.8.3 Window resize.....	268
12.8.4 El evento scroll.....	268
12.9 PRACTICA Y JUEGA.....	269
CAPÍTULO 13. EL DOM DE JAVASCRIPT.....	270
13.1 PROCESO DE CARGA	270
13.2 LOS NODOS Y SUS TIPOS	271
13.3 SELECCIÓN DE ELEMENTOS	271
13.3.1 Interfaz NodeList.....	272
13.3.2 Los selectores	273
13.3.3 Métodos para acceder a los nodos y elementos	273
13.4 MANIPULACIÓN DE NODOS Y ELEMENTOS.....	274
13.4.1 Interfaz DOMTokenList	274
13.4.2 Método createElement	275

13.4.3	Propiedad id	275
13.4.4	Propiedad innerHTML	276
13.4.5	Propiedad value	276
13.4.6	Método setAttribute	276
13.4.7	Propiedad classList	277
13.4.8	Propiedad previousElementSibling	277
13.4.9	Propiedad nextElementSibling	278
13.4.10	Propiedad parentElement	278
13.4.11	Método appendChild	278
13.4.12	Método insertBefore	278
13.5	ELIMINACIÓN DE NODOS Y ELEMENTOS	280
13.5.1	Método remove	280
13.5.2	Método removeChild	280
13.6	DEFINICIÓN DE ESTILOS	281
13.6.1	La interfaz CSSStyleDeclaration	281
13.6.2	La interfaz CSSStyleSheet	282
13.6.3	Propiedad style	284
13.6.4	Método insertRule	284
13.6.5	Método deleteRule	285
13.7	PRACTICA Y JUEGA	285
CAPÍTULO 14. JAVASCRIPT ASÍNCRONO		286
14.1	EL ESTÁNDAR CORS	286
14.1.1	Encabezados de solicitud HTTP	287
14.1.2	Encabezados de respuesta HTTP	287
14.2	CONEXIONES HTTP	288
14.2.1	Objeto XMLHttpRequest	289
14.2.2	Eventos	292
14.2.3	Ejemplo sencillo de XMLHttpRequest	293
14.3	PROMESAS	293
14.3.1	Objeto Promise	293
14.3.2	La API fetch	297
14.4	PRACTICA Y JUEGA	301
CAPÍTULO 15. VALIDACIÓN DE FORMULARIOS CON JAVASCRIPT		302
15.1	PROPIEDADES DE LOS FORMULARIOS	302
15.2	PROPIEDADES DE LOS ELEMENTOS DE FORMULARIO	304
15.3	CREACIÓN Y ENVÍO DE FORMULARIOS	305
15.4	VALIDACIÓN DE FORMULARIOS	306
15.4.1	La interfaz ValidityState	306
15.4.2	Propiedades y métodos	307
15.4.3	Eventos	308
15.4.4	Ejemplo de validación	309
15.5	PRACTICA Y JUEGA	310
REFERENCIAS		311
MATERIAL ADICIONAL		313

GLOSARIO DE TÉRMINOS

A continuación, se explican algunos conceptos previos que se deben tener antes de empezar con el curso.

Concepto	Descripción
Accesibilidad Web	La accesibilidad web es una de las partes que engloba la usabilidad web. Mientras que la usabilidad web, entre otras cosas, se centra en el rendimiento, la semántica y la universalidad, la accesibilidad web, se asegura de que todo tipo de usuario, tenga o no una limitación, discapacidad o incapacidad, puedan usar las páginas y aplicaciones con una experiencia de usuario óptima.
Agente de usuario	Es un programa informático o software que funciona como interfaz de interacción web o cliente de red. Habitualmente, este tipo de software hace referencia a los navegadores web y herramientas de asistencia a la accesibilidad como pueden ser los magnificadores de pantalla, los lectores de pantalla, etcétera.
Diseño Gráfico	Podría definirse como una disciplina que consiste en presentar información visual y cuyo objetivo es que, los usuarios, capten mensajes específicos sobre un tema o materia determinada.
IDE	IDE es un acrónimo que significa Entorno de Desarrollo Integrado. Su función, es la de mejorar la productividad de los desarrolladores valiéndose de unas herramientas para la edición de código, de construcción automática que evitan, entre otras cosas, errores de sintaxis y de estructuración y, un depurador. También pueden contener un compilador y/o intérprete como es el caso de NetBeans o Eclipse.
Lenguaje de marcado	Un lenguaje de marcado es un mecanismo para codificar documentos con todo tipo de objetos, sean textuales o no textuales. Esta codificación, a menudo, permite el uso de etiquetas y atributos para proporcionar información adicional a la estructura del texto o su presentación.

Metadato	Un metadato puede definirse como información adicional acerca de los propios datos. Muchos autores lo definen como datos sobre los datos, que es una forma de decir el contexto de los datos o a qué se refieren esos datos. Normalmente, el identificador del metadato ya indica su objetivo y, su valor, indica la explicación o contextualización.
Microdato	Los microdatos son unas marcas adicionales que se emplean para anidar metadatos sobre una información concreta dentro de un documento basado en lenguaje de marcado.
Polyfill	Un polyfill es un componente o fragmento de código que habilita o posibilita una funcionalidad que el agente de usuario no es capaz de proporcionar, ya sea porque es antiguo, ya sea porque utiliza un estándar incompatible.
Mobile First	Mobile First es una disciplina que considera que las páginas se deben diseñar dando prioridad a los dispositivos móviles, es decir, primero se tiene en cuenta el diseño para un dispositivo móvil y, si el escenario lo permite o es diferente, se le aplican una serie de reglas para su adecuado funcionamiento y correcta visualización.
Seguridad Web	La seguridad web se refiere cómo se debe contemplar o proveer la seguridad de la información en las páginas y aplicaciones, sean del tipo que sean. Su objetivo es evitar ataques de usuarios malintencionados y de softwares diseñados con un propósito no lícito o ético. En otras palabras, es una disciplina que permite proteger contra el acceso, manipulación, destrucción e interrupción de información no autorizado.
Semántica Web	La semántica web es un conjunto de recomendaciones y estándares desarrolladas por la W3C (World Wide Web Consortium) que están pensadas para hacer que los datos se vuelvan más legibles.
Usabilidad Web	En términos generales, la usabilidad se refiere a la facilidad con la que las personas utilizan los programas y máquinas que deben de manejar. En el campo de la informática, la usabilidad se refiere a la facilidad con la que las personas manejan las páginas y aplicaciones, sean web o móviles. Por tanto, un diseño usable es aquel que está centrado en el usuario.

1

CONCEPTOS PREVIOS

1.1 ENTORNOS DE PROGRAMACIÓN

Un entorno de programación o entorno de desarrollo integrado (IDE) es un programa o aplicación de software que proporciona soluciones integrales a los programadores para desarrollar software, sea del tipo que sea.

Todos los IDE normalmente poseen un editor de código fuente, herramientas varias para la automatización de tareas, validadores de código como linters, opciones para la compilación y un depurador, entre otras características.

Uno de los IDE más extendidos para múltiples arquitecturas como puedan ser PHP, Angular, React, JavaScript, HTML y CSS es Visual Studio Code.

Visual Code es un IDE (Entorno de Desarrollo Integrado) desarrollado por Microsoft que permite editar los archivos en modo texto y presenta multitud de extensiones entre las que se incluyen soporte para la depuración, control integrado de Git y el resaltado de sintaxis.

Es descargable desde:

<https://code.visualstudio.com/download>

1.2 HISTORIA DE LA WEB



En los años 60 se produjo una nueva forma de compartir información con otros usuarios. Se trataba de un servicio de comunicación que sólo permitía la inclusión de textos y eran manipulables a través de navegadores de sólo texto. Sin embargo, no fue hasta principios de los noventa cuando se creó HTML, lo que provocó que la web empezara a tener una aceptación suficiente y extenderse como la pólvora.

En aquel entonces, la web era un sistema unidireccional de publicación estático de sólo texto que no presentaba gráficos o imágenes, no ofrecía opciones de personalización, no permitía la actualización y, mucho menos, la posibilidad de realizar intercambio de datos, por lo que los usuarios no podían interactuar con el contenido y, únicamente, se limitaban a consultar o leer la información que el administrador de la página web hubiese subido a la red. A esto, se denominó la Web 1.0.

La web 2.0, término que fue bautizado por O'Reilly en el año 2004, supuso la segunda generación de sitios y páginas web. Este tipo de webs ya no eran estáticas, ni de sólo lectura y permitían, entre otras cosas, compartir e interactuar con la información de una manera sencilla. Como consecuencia de ello, se produjo un desarrollo de la inteligencia colectiva que fomentaba la colaboración y el intercambio de información a través de comunidades y redes sociales.

Es, por esta época, cuando se crean y extienden sistemas tan conocidos como son los blogs, chats, wikis o foros. Sistemas bidireccionales, los cuales, permitían manipular y gestionar la información de forma sencilla, además de permitir la adición de comentarios

y opiniones o interactuar con otros usuarios que presentaban las mismas inquietudes, pero que no querían tener el mismo nivel técnico o cultural.

Sin embargo, no fue hasta la web 3.0 dónde se produjo un salto cuantioso en lo referente a los sistemas en red. La web 3.0, la tercera generación de sitios y páginas web, ya no sólo era una forma de interactuar y compartir la información de manera sencilla, ahora, su objetivo es darle significado y enriquecer la experiencia del usuario.

Es aquí, como alguno ya habrá pensado, cuando nace la Semántica web y las páginas web empiezan a estructurarse a través de un lenguaje natural que puede ser interpretado por el software definiendo qué parte tiene cada función. De esta forma, acceder a la información resulta más sencillo y rápido de procesar, porque hasta las máquinas son capaces de “entender” los contenidos y su objetivo.

Se dice que la web 3.0 también tiene bastante que ver con la inteligencia artificial puesto que las páginas y aplicaciones web ya poseen la capacidad de conectarse entre sí para ofrecer un mejor servicio a los intereses de cada usuario. No obstante, es en la web 4.0 dónde esta premisa está más presente puesto que es quién obtiene un comportamiento más inteligente, predictivo y simple que implica menos movimientos y más acciones con mejores resultados.

Con la web 4.0, nacen el aprendizaje profundo (Deep Learning) y el aprendizaje automático (Machine Learning), tecnologías que forman parte de una familia de métodos de aprendizaje automático basados en redes neuronales con aprendizaje de representación. En otras palabras, tecnologías basadas en sistemas capaces de aprender a realizar tareas tras analizar diferentes patrones y muestras mediante técnicas de aprendizaje que permiten descubrir de manera automática las características de una entidad a partir de datos sin procesar.

El ejemplo más conocido o extendido de todo esto quizás sea Watson de IBM, un software capaz de responder preguntas realizadas en lenguaje natural y de realizar tareas como Speech To Text, el cual permite hablarle a una máquina y convertir lo dicho en texto escrito.

Pero esto no es todo, la web 4.0 es la responsable de que los usuarios sean advertidos por sus dispositivos móviles antes de que ellos mismos se den cuenta. Por ejemplo, ¿quién no ha recibido notificaciones con la ruta más corta al trabajo, avisos por atascos en la carretera, alertas por fuertes tormentas o lluvias o mensajes de advertencia sobre tu elevado ritmo cardíaco?

1.3 CÓMO FUNCIONA LA WEB

De forma básica, cuando un usuario se conecta a Internet con un dispositivo cualquiera, se le asigna un identificador único mediante los protocolos TCP/IP (Protocolo de Control de Transmisión / Protocolo de Internet). El protocolo TCP proporciona el medio para crear las conexiones y el protocolo IP proporciona el mejor “camino” para alcanzar su destino.

Este identificador único, más conocido como dirección IP, suele estar compuesto por cuatro códigos de 8 bits y vinculado a un nombre, también único, el cual utilizamos para acceder a un sitio web (véase, por ejemplo, *https://google.es*).

¿Qué es lo que sucede entre medias? Como hemos dicho, Internet se mueve a través de direcciones IP, por lo que, para conseguir la dirección IP asignada a ese nombre que hemos introducido, primero se debe acceder a un sistema intermedio que almacena dicha relación.

Ese sistema intermedio se conoce como DNS (Sistema de Nombres de Dominio) y, fundamentalmente, lo que hace es recopilar un catálogo de correspondencias de nombres e IPs y devolver un valor concreto como, por ejemplo, 216.58.211.35.

Una vez que se tiene el objetivo al que dirigirse, el navegador, también llamado Cliente en términos de comunicaciones, abre una instancia de comunicación con el Servidor mediante el protocolo HTTP (Protocolo de Transferencia de Hipertexto). Este protocolo es quién dicta las normas para que el Cliente se comunique con el Servidor Web asignado a la IP anteriormente adquirida y es, además, quién define la sintaxis y semántica que se debe utilizar en cada conexión.

No obstante, si accedemos a la consola del navegador (pulsando F12) y recuperamos la información de la pestaña NETWORK, al recargar la página veremos que la mayoría de estas conexiones entre el Cliente y el Servidor se realizan a través de HTTPS, o lo que es lo mismo, la versión segura del protocolo HTTP.

En este tipo de comunicación, el servidor establece un cifrado basado en la seguridad de textos mediante los protocolos criptográficos SSL/TLS, los cuales, permiten crear una capa codificada intermedia entre los protocolos HTTP y TCP/IP por el que envía el código HTML que el navegador muestra al usuario.

A continuación, se muestra un gráfico que representa todo el proceso:

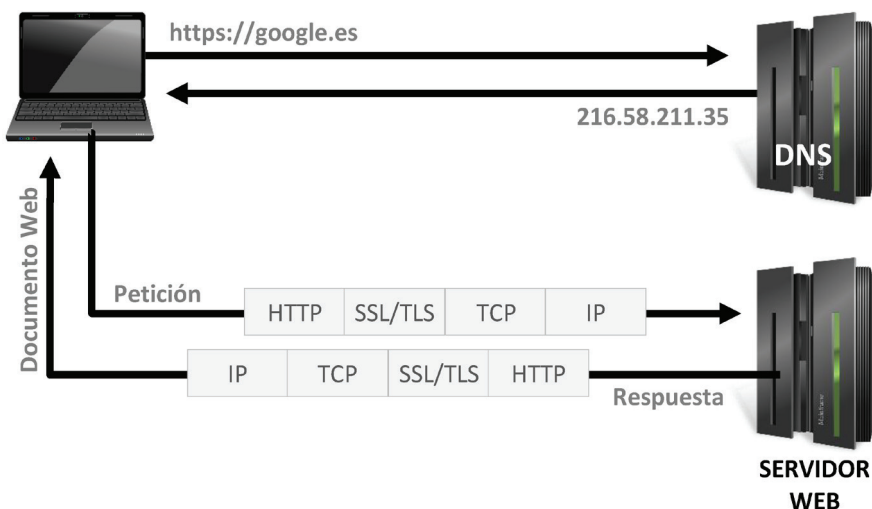


Ilustración 1.3. Proceso de una petición web.

1.4 DISEÑO GRÁFICO Y EL PAPEL DEL DISEÑADOR

El diseño gráfico podría definirse como una disciplina que consiste en presentar información visual y cuyo objetivo es que, los usuarios, capten mensajes específicos sobre un tema o materia determinada.

También podríamos definir el diseño gráfico como el arte de crear una composición de imágenes y texto de forma que se transmitan mensajes de forma efectiva.

Cuando hablamos de diseñadores gráficos, no estamos hablando de comerciales ni de artistas, no obstante, es un perfil que tiene algo de ambos. El objetivo de un diseñador gráfico no es vender nada, aunque pueda hacerlo. Ni tampoco es ser un experto comunicador, aunque puede y debe tener una cierta maestranza para poder abstraer los mensajes que se quieren divulgar y transmitirlos de forma eficiente.

Por tanto, lo primero que necesita un diseñador gráfico para empezar a trabajar es un documento o reunión informativa en la que se le proporcione todos los datos necesarios para poder afrontar de manera ordenada, estratégica y creativa el objetivo a cumplir.

Más tarde, cuando ya disponga de los requerimientos y necesidades, y tenga una idea más o menos clara de lo que se debe crear, el diseñador podrá ayudarse, y en general lo hará, de herramientas específicas como son PhotoShop, Illustrator, Gimp o FireWorks, y de lenguajes de marcado como HTML5, CSS3 o SVG, los cuales facilitan el trabajo en gran medida.

1.5 ARQUITECTURA DE LA INFORMACIÓN Y EL PAPEL DEL ARQUITECTO

La Arquitectura de la Información (AI) es el arte, disciplina o metodología, según quien lo defina, encargada de estudiar, analizar, organizar, y estructurar la información.

Cuando se habla de Arquitectura de la Información, a lo que se hace referencia es a unos principios de diseño y construcción que ayudan a seleccionar y presentar la información en sistemas interactivos y pasivos.

En lo referente a lo que busca la Arquitectura de la Información podríamos decir que, su objetivo, es:

- Procesar y dosificar la ingente cantidad de información que se produce cada día en las redes.
- Desarrollar y verificar los procesos de producción de forma que los usuarios puedan acceder con los mínimos problemas posibles.
- Utilizar la información de manera clara y precisa.
- Organizar, estructurar, marcar y distribuir los sistemas de información para que la experiencia de los usuarios se vuelva más sencilla y menos frustrante.

Si pensamos en la Arquitectura de la Información como proceso, lo que encontramos es una especie de ecosistema que tiene un ciclo de vida completo y que va definiendo los objetivos para que la producción y desarrollo se realice de forma efectiva y eficiente.

De hecho, para que los contenidos puedan ser accedidos, utilizados y asimilados de manera eficiente y efectiva, la Arquitectura de la Información define una serie de técnicas que, dependiendo de cuáles sean y de cuándo se produzcan, ayudan al desarrollo y creación de sistemas de información.

Estas técnicas pueden definir:

- El objeto o propósito del producto.
- El público objetivo.
- Los estudios de audiencia y análisis de competencia.
- La planificación, gestión y desarrollo de contenidos.
- El diseño de la interacción, la Usabilidad y la Accesibilidad.
- El diseño de la navegación, organización y apariencia de los contenidos.
- El marcado de los datos para acceder a los contenidos y facilitar su búsqueda.
- Los procesos de reingeniería del sistema.

Por tanto, un arquitecto de la Información es una persona que lleva a cabo la ejecución y verificación del diseño del sistema o interfaz, además de estar en constante cooperación con los diseñadores gráficos y responsables de la parte de procesamiento y lógica de negocio para definirla.

Sin embargo, no me sentiría cómodo al cerrar esta parte, sin antes, realizar unos últimos comentarios. Cuando se habla de Experiencia de Usuario (UX) se puede estar haciendo referencia a la Arquitectura de la Información, porque es una parte o componente de misma. No obstante, mientras que la Arquitectura de la Información se centra en la información, la Experiencia de Usuario se centra en los usuarios y en cómo pueden actuar y pensar.

Piénsese que el concepto de UX es un conglomerado de técnicas y elementos que define o ayuda a definir la estructura y organización de la información, el diseño de interacción, la usabilidad y accesibilidad de los sistemas y que, para ello, se apoya en el diseño gráfico y procesos cognitivos y de percepción.

1.6 ESTRUCTURA BÁSICA DE UNA PÁGINA WEB

Todo documento web presenta siempre una misma estructura básica. Una cabecera que prepara el documento, un contenido central o principal que habitualmente se

denomina cuerpo del documento y un pie de página que suele proporcionar algunos contenidos adicionales.

La cabecera de la página es la parte en donde se proporcionan todos los datos de configuración del documento y todos los datos que describen el contenido del documento que se va a presentar. Estos datos que describen el contenido del documento se denominan formalmente metadatos y, en pocas palabras, se definen como datos acerca de los datos.

Estos metadatos pueden ser de muy diferente índole, desde datos puramente informativos, como quién creó el documento o el título del mismo, hasta datos que configuran la presentación del contenido. Estos últimos, pueden preparar la codificación de caracteres a usar en el documento, establecer las palabras clave que permiten la indexación en los motores de búsqueda como Google o Yahoo!, permitir o no que la página sea rastreada por los robots de los motores de búsqueda, definir el periodo de validez de los datos en caché, etcétera, etcétera, etcétera.

El cuerpo de la página, por el contrario, no describe nada referente al propio documento, sino que proporciona todo el contenido perceptible, es decir, es la parte del documento donde se establecen los textos, imágenes, vídeos, gráficos estadísticos e, incluso, los sonidos o música de fondo.



Ilustración 1.4.1. Ejemplo de estructura básica de una página web.

Como se puede observar en la ilustración anterior, cada elemento de una página web suele estar vinculado a una sección determinada. Por ejemplo, el logo y menú principal suelen estar ubicados en la zona de la cabecera y, en el pie de página, se suele mostrar información sobre el copyright.

No obstante, las secciones, casi nunca suelen tener tan pocos elementos o ser tan triviales. De hecho, los pies de página, habitualmente, suelen llevar otros contenidos adicionales como puedan ser los enlaces de interés, los créditos, acceso a las redes sociales y, en algunos casos proporcionan un formulario para apuntarse a la Newsletter o para realizar un contacto.

1.7 LOS ESTÁNDARES W3C

Los estándares Web W3C (World Wide Web Consortium) son un conjunto de recomendaciones internacionales creadas con la finalidad de conseguir que los sitios web obtengan un mejor rendimiento y mejorar la funcionalidad operativa tanto para los navegadores como para los usuarios.

El W3C desarrolla estándares web internacionales para HTML, CSS y semántica web, entre otros muchos y ofrecen soporte para la accesibilidad a nivel mundial. Este último es revisado por el Grupo de Trabajo de Arquitecturas de Plataforma Accesible - APA (en inglés).

Entre los más relevantes tenemos las Pautas de accesibilidad para el contenido web (WCAG), las Pautas de accesibilidad para las herramientas de creación de contenido (ATAG) y las Pautas de accesibilidad para el agente de usuario (UAAG).

Si se desea más información se puede visitar la dirección:



<https://www.w3.org/WAI/standards-guidelines/es>

2

INTRODUCCIÓN A HTML5

2.1 INTRODUCCIÓN

El lenguaje HTML (HyperText Markup Language o lenguaje de marcado de hipertexto) es un lenguaje de marcado dedicado a la elaboración de páginas web. Fue definido por primera vez en 1991 y, en aquel entonces, se caracterizaba por tener algo más de una docena de etiquetas. Más tarde, en 1995 se publicó el primer estándar oficial de HTML al que denominaron HTML 2.0.

En 1997 entró en juego la W3C y desarrolló tres estándares más hasta llegar a lo que hoy conocemos como HTML5 en 2014.

Si bien HTML es un lenguaje formado por entidades que ayudan a estructurar y proporcionar significado a las diferentes partes del documento, cada una de estas entidades, usualmente denominadas elementos o etiquetas, están formadas por un contenido y cero, uno o varios atributos.

```
<p>Esto es un párrafo</p>
<div class="layer">Esto es una capa</div>
```

Cada uno de los atributos tiene una función y puede estar o no asociado a un comportamiento o definición específica. Por ejemplo, el atributo ID habitualmente es utilizado para poder manipular el elemento a través de un nombre corto, sin embargo, también puede ser declarado para vincularse con otro elemento generando una entidad mayor, como es el caso del siguiente código.

```
<label for="nombre">Nombre</label>
<input id="nombre" placeholder="Inserte el nombre completo" />
```

Ilustración 2.2. Etiquetado de un campo de formulario en HTML

El atributo FOR, utiliza el atributo ID para vincular el LABEL con el INPUT y generar un elemento combinado o pequeño componente.

Cabe destacar que, aunque puede haber etiquetas sin cierre, como es el caso del elemento INPUT, lo normal es que todas las etiquetas o marcas tengan un principio y un final, como es el caso de la etiqueta LABEL.

En lo referente a las novedades de HTML5, como muchos sabrán, una de las más significativas es el valor semántico. La semántica es una característica que dota a los documentos web de mayor significado porque, entre otras cosas, proporciona una mayor estructuración y ayuda a la compresión gracias a lo que se denomina identificador semántico.

El identificador semántico es un término que hace referencia a lo que contiene o representa la etiqueta, es decir, cada etiqueta o elemento tiene un nombre asociado que representa o indica su objetivo. Por ejemplo, en general, la etiqueta SECTION siempre contendrá un conjunto de elementos agrupados que tendrán o guardarán una relación.

2.2 ELEMENTOS BÁSICOS DE HTML

2.2.1 Definición del tipo de documento DTD (!DOCTYPE)

Cuando uno decide trabajar con HTML, lo primero que debe hacer es declarar el elemento !DOCTYPE. Este elemento tiene, como objetivo, informar al navegador del tipo de documento que se va a definir.

La Declaración del Tipo de Documento (DTD) puede cambiar, y de hecho cambia, para cada versión de HTML. La versión del lenguaje de marcado puede ser muy diferente según qué tipo se utilice, y puede tener más o menos restricciones en función del modo y versión. Sin ir más lejos, el tipo de documento que se debe definir para indicar que es un documento XHTML es muy distinto al que se debe usar para indicar que es HTML5 o SVG.

A continuación, se muestran los principales DTD para documentos de HTML, SVG y MathML.

2.2.1.1 DTDS DE HTML

HTML5

```
<!DOCTYPE html>
```

2.2.1.2 DTDS DE MATHML

MathML 2.0

```
<!DOCTYPE math PUBLIC "-//W3C//DTD MathML 2.0//EN"
    "http://www.w3.org/TR/MathML2/dtd/mathml2.dtd">
```

2.2.1.3 DTDS DE SVG

SVG 1.1 Full

```
<!DOCTYPE svg PUBLIC
    "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

SVG 1.1 Básico

```
<!DOCTYPE svg PUBLIC
    "-//W3C//DTD SVG 1.1 Basic//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-basic.dtd">
```

SVG 1.1 Reducido

```
<!DOCTYPE svg PUBLIC
    "-//W3C//DTD SVG 1.1 Tiny//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-tiny.dtd">
```

2.2.1.4 DTDS DE XHTML

XHTML1.1

```
<!DOCTYPE html PUBLIC
    "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

2.2.2 Etiqueta html

La etiqueta HTML es el elemento que representa la raíz o base de un documento HTML y supone el cierre automático del resto de los elementos declarados posteriormente a él.

La etiqueta HTML admite varios atributos, la mayoría en desuso. El único que sigue estando vigente es LANG y es el encargado de definir el lenguaje del documento.

Es un atributo muy útil cuando se dispone de documentos en distintos idiomas y para aquellos usuarios que dependen de herramientas de asistencia como lectores de pantalla.

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

2.2.3 Etiqueta head

La etiqueta HEAD es el elemento o la estructura que proporciona información general acerca del documento. Esta información general viene definida a modo de metadatos, o lo que es lo mismo, datos que informan sobre los datos y, pueden ser de muy diferente índole. Esto es, el tipo de codificación, el título del documento, las palabras clave que lo describen, la descripción sobre lo que contiene, el autor del documento, etcétera.

No obstante, lo que más abunda dentro de esta estructura suele ser elementos LINK o STYLE, los cuales recogen todas las reglas CSS aplicables en el documento.

```
<head>
  <!-- Información del documento -->
</head>
```

NOTA

Aunque la etiqueta SCRIPT puede estar definida dentro del elemento HEAD, lo mejor es que esté al final de la etiqueta BODY para evitar bloqueos o retrasos en la muestra del primer renderizado.

2.2.4 Etiqueta body

La etiqueta BODY es el elemento o la estructura que define todo el contenido útil del documento. Aquí es dónde se definirán todos los textos, capas, botones, controles de entrada y salida, etcétera para que los usuarios puedan utilizarlo o consultarlo.

Al final de esta estructura, habitualmente, suele contener uno o varios elementos SCRIPT que todas las funcionalidades que se ejecutan en el navegador, como validaciones o animaciones.

```
<body>
  <!-- Contenido del cuerpo de la página -->
</body>
```

2.2.5 Comentarios

Los comentarios en HTML se establecen a través de las marcas <!-- y -->. Estas etiquetas o marcas indican al navegador que la información contenida no debe ser interpretada y, por tanto, tampoco renderizada. Un ejemplo podría ser:

```
<!-- Esto es un comentario de HTML -->
```

2.3 INFORMACIÓN DEL DOCUMENTO

Los elementos que proporcionan información sobre el documento son las etiquetas `BASE`, `LINK`, `META`, `SCRIPT`, `STYLE` y `TITLE`.

2.3.1 Elemento base

El elemento `BASE` especifica la base de direccionamiento predeterminada para los elementos que utilicen un direccionamiento relativo, es decir, las direcciones no tienen como primer carácter el símbolo de barra inclinada (`/`) y no empiezan por un identificador de dominio.

Debe estar dentro del elemento `HEAD` y es importante destacar que, si se declaran varios elementos `BASE`, la última declaración anulará todas las anteriores.

```
<head>
  <base href="https://www.ejemplo.com/" target="_blank" />
</head>
```

2.3.2 Elemento link

El elemento `LINK` especifica un enlace hacia una hoja de estilos o archivo externo.

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

El único atributo obligatorio del elemento `LINK` es `REL`, que es quién define la relación que existe entre el documento actual y el enlazado, es decir, si es una hoja de estilos, un icono, un archivo de ayuda, un documento de alternativa, ...) y su valor más recurrente es `STYLESHEET`.

No obstante, aunque el establecimiento de este atributo es importante y ayuda a la semántica web, también puede influir en el rendimiento de forma notable haciendo que los recursos se carguen de maneras diferentes.

Por esta razón, es importante que se conozca bien la aplicación que se está creando y optimizar los recursos, porque un mal uso del atributo `REL` puede influir negativamente en el rendimiento de la página.

A continuación, se muestran algunos de los valores de `REL` que afectan al rendimiento:

Valor	Descripción y ejemplo
length	<p>La cláusula DNS-PREFETCH indica al agente de usuario que se resuelva el DNS lo antes posible, es decir, que se resuelva el dominio del servidor, pero no realice ninguna descarga.</p> <p>Es especialmente útil cuando se desean cargar archivos desde fuentes externas como pueda ser una fuente vectorial de Google Fonts, un script de un CDN o un JSON desde una API.</p> <pre><link rel="dns-prefetch" href="//fonts.googleapis.com"/></pre>
prefetch / preload	<p>Las cláusulas PREFETCH y PRELOAD indican al agente de usuario que descargue y almacene en caché un recurso determinado, como pueda ser una hoja de estilos o un script.</p> <p>Mientras que la cláusula es PREFETCH provocará que la descarga se realice con prioridad baja, es decir, sin afectar a los recursos más importantes o prioritarios, la cláusula es PRELOAD provocará que la descarga se realice a la mayor brevedad posible, pudiendo afectar a los recursos más importantes o prioritarios.</p> <p>Ambas cláusulas se alimentan del atributo AS, que proporciona una pista de la prioridad del recurso al agente de usuario. Sus principales valores son STYLE, SCRIPT, FONT e IMAGE, aunque hay más.</p> <pre><link rel="preload" href="/js/scripts.js" as="script"></pre>
preconnect	<p>La cláusula PRECONNECT indica al agente de usuario que resuelva el DNS y realice la preconexión con los protocolos TCP y TLS, si procede.</p> <p>Al igual que DNS-PREFETCH, es especialmente útil cuando se desean cargar archivos desde fuentes externas como pueda ser una fuente vectorial de Google Fonts, un script de un CDN o un JSON desde una API. Sin embargo, su uso excesivo puede provocar pérdidas sustanciales en el rendimiento global, por lo que no se recomienda usarlo más de 4 o 6 veces por página.</p> <pre><link rel="preconnect" href="//islavisual.com/"></pre>
prerender	<p>La cláusula PRERENDER indica al agente de usuario que se renderice el recurso en segundo plano. Esto puede ser una buena idea cuándo se está seguro de que, el usuario, realizará alguna acción que requiera esta precarga.</p> <p>Su uso puede aumentar el rendimiento hasta un 70%, no obstante, este tipo de procesamiento es muy costoso, tanto a nivel de memoria, como a nivel de tráfico</p> <p>Cabe destacar que, a febrero de 2023, esta cláusula no está del todo soportada por Firefox ni Safari, por lo que su uso no está recomendado.</p> <pre><link rel="preconnect" href="//islavisual.com/"></pre>

Con respecto a su posible personalización, permite, entre otras cosas, establecer el idioma en el que está escrito a través del atributo `HREFLANG` y el dispositivo o medio para el que está optimizado mediante el atributo `MEDIA`, que habitualmente es `ALL`, `SCREEN` o `PRINT`.

2.3.3 Elemento meta

Los metadatos se pueden definir como datos acerca de los datos. Es como una información que nos proporciona datos clave sobre el contenido que va a ser representado.

Por tanto, podríamos decir que los metadatos proporcionan información a los robots y usuarios que lo necesiten sobre el documento actual. No obstante, esta información puede ser de muy diversa índole, desde información referente a la apariencia, hasta datos que pueden ser utilizados por los agentes de usuario.

Dicho esto, el elemento `META` especifica una información sobre los datos que se encuentran dentro del actual documento HTML. Un ejemplo podría ser:

```
<meta name="keywords" content="HTML5, HTML, XHTML">
```

El elemento `META` se alimenta de varios atributos que permiten personalizar, entre otras cosas, la codificación de caracteres, respuestas al documento o el esquema que se debe emplear, lo cual pasamos a ver a continuación.

2.3.3.1 ATRIBUTOS DEL ELEMENTO META

2.3.3.1.1 Atributo name

El atributo `NAME` especifica el nombre clave que va a asignarse al metadato y asociarse a su par nombre-valor determinado por el atributo `CONTENT`.

En general, podemos decir que, el atributo `NAME` permite definir el tipo de metadato que se va a especificar. Esto es, por ejemplo, una descripción, un autor, las palabras clave (`KEYWORDS`) que clasifican el documento o, el tamaño y escalamiento de la ventana gráfica.

```
<meta name="author" content="Pablo E. Fernández Casado">
```

2.3.3.1.2 El atributo content

El atributo `CONTENT` especifica el valor de los metadatos referidos a la página o a directivas pragma.

El atributo de `CONTENT` sólo debe establecerse si se define el atributo `NAME` o el atributo `HTTP-EQUIV` por lo que, si el metadato que se está definiendo no contiene ninguno de estos atributos, se debe obviar su declaración para evitar errores de procesamiento, accesibilidad y/o usabilidad web.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

2.3.3.1.3 El atributo charset

El atributo `CHARSET` se utiliza para especificar la codificación de caracteres en la que viene definido el documento. Entre sus valores más recurrentes podemos encontrar el valor UTF-8 y el valor ISO-8859-1.

El atributo `CHARSET` sólo puede ser interpretado una única vez por documento, es decir, que debe ser declarado una única vez en la página. Además, debe estar declarado al principio del documento, entre los primeros 512 bytes, dentro de la etiqueta `HEAD` porque, en caso contrario, puede ser ignorado.

Cabe destacar que, aunque existen multitud de posibles codificaciones, ningún agente de usuario las soporta todas, por lo que es importante que uno se informe antes de establecer una codificación de caracteres específica.

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

2.3.3.1.4 El atributo http-equiv

El atributo `HTTP-EQUIV` se dice que es una directiva pragma y vienen a ser simulaciones de encabezados de respuesta HTTP. En otras palabras, se utiliza para comunicarse con los servidores y adaptar sus respuestas al documento.

Este atributo era utilizado antes de la versión 5 de HTML para especificar la codificación de caracteres, pero ahora ya no está permitido. A continuación, se muestra una lista con los valores actuales más representativos que admite este atributo.

- **CONTENT-SECURITY-POLICY:** especifica una política de contenido que sirve para prevenir y disminuir algunos tipos de ataque como la inyección de datos o XSS (Cross Site Scripting) o para definir algún comportamiento que, se desea, se cumpla. Por ejemplo, si se desea que todo el contenido provenga del mismo origen, se puede establecer esta directiva a `DEFAULT-SRC 'SELF'`. Si, además, se desea que incluya todos sus subdominios, podría definirse la directiva a `DEFAULT-SRC 'SELF' *.COMPONENTS.COM`. (Para más información puede visitarse la URL <https://developer.mozilla.org/es/docs/Web/HTTP/CSP>).
- **DEFAULT-STYLE:** especifica la hoja de estilos por defecto o preferida. El valor de `CONTENT` debe ser exactamente el mismo que el definido en el atributo `TITLE` del elemento `LINK` o elemento `STYLE`.
- **REFRESH:** especifica un intervalo de tiempo, tras el cual, el documento se actualiza automáticamente.

```
<meta http-equiv="refresh" content="60">
```

2.3.3.1.5 El atributo scheme

El atributo `SCHEME` *no está soportado por HTML5*, no obstante, antes se utilizaba para especificar el esquema que se debía emplear para interpretar el valor de una propiedad.