Iniciación a la creación de páginas web

Iniciación a la creación de páginas web

Pablo E. Fernandez Casado

Ra-Ma®

Iniciación a la creación de páginas web
Thema: UGB Gráficos y diseño web
Bisac: COM062000
© Pablo E. Fernandez Casado
© De la edición: Ra-Ma 2024

MARCAS COMERCIALES. Las designaciones utilizadas por las empresas para distinguir sus productos (hardware, software, sistemas operativos, etc.) suelen ser marcas registradas. RA-MA ha intentado a lo largo de este libro distinguir las marcas comerciales de los términos descriptivos, siguiendo el estilo que utiliza el fabricante, sin intención de infringir la marca y solo en beneficio del propietario de la misma. Los datos de los ejemplos y pantallas son ficticios a no ser que se especifique lo contrario.

RA-MA es marca comercial registrada.

Se ha puesto el máximo empeño en ofrecer al lector una información completa y precisa. Sin embargo, RA-MA Editorial no asume ninguna responsabilidad derivada de su uso ni tampoco de cualquier violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Esta publicación tiene por objeto proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para el editor ninguna forma de asistencia legal, administrativa o de ningún otro tipo. En caso de precisarse asesoría legal u otra forma de ayuda experta, deben buscarse los servicios de un profesional competente.

Reservados todos los derechos de publicación en cualquier idioma.

Según lo dispuesto en el Código Penal vigente, ninguna parte de este libro puede ser reproducida, grabada en sistema de almacenamiento o transmitida en forma alguna ni por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro sin autorización previa y por escrito de RA-MA; su contenido está protegido por la ley vigente, que establece penas de prisión y/o multas a quienes, intencionadamente, reprodujeren o plagiaren, en todo o en parte, una obra literaria, artística o científica.

Editado por:
RA-MA Editorial
Calle Jarama, 3A, Polígono Industrial Igarsa
28860 PARACUELLOS DE JARAMA, Madrid
Teléfono: 91 658 42 80
Fax: 91 662 81 39
Correo electrónico: *info@grupoeditorialrama.com*
Internet: *www.ra-ma.es* y *www.ra-ma.com*
ISBN impreso: 978-84-10181-81-6
ISBN ePub: 978-84-10181-82-3
Depósito legal: M-8637-2024
Maquetación: Antonio García Tomé
Diseño de portada: Antonio García Tomé
Filmación e impresión: Safekat
Impreso en España en marzo de 2024

La ciencia más útil
es aquella cuyo fruto
es el más comunicable

ÍNDICE

SOBRE EL AUTOR **11**

GLOSARIO DE TÉRMINOS **13**

CAPÍTULO 1. INTRODUCCIÓN A INTERNET **15**

1.1 INTRODUCCIÓN A INTERNET 15
1.2 NAVEGADORES DE INTERNET 16
1.3 EL CORREO ELECTRÓNICO EN INTERNET 18
1.4 OTRAS APLICACIONES DE INTERNET 19
1.5 HERRAMIENTAS DE DISEÑO DE PÁGINAS WEB 21

CAPÍTULO 2. PLANIFICACIÓN DE LA WEB **23**

2.1 DISEÑO DEL CONTENIDO 23
2.2 PROPÓSITO Y AUDIENCIA DE LA WEB 24
2.2.1 Propósito de la web 24
2.2.2 Audiencia de la web 25
2.3 DISEÑO DE LA APARIENCIA 26
2.4 CONSTRUIR EL SITIO 27
2.5 ANALIZAR LA ESTRUCTURA 29
2.6 ORGANIZAR ARCHIVOS 30

CAPÍTULO 3. LENGUAJE HTML **33**

3.1 PROGRAMACIÓN EN HTML 33
3.2 CREACIÓN DE UNA PÁGINA EN HTML 34

3.2.1 Definición del tipo de documento DTD (!DOCTYPE) ... 34
3.2.2 Etiqueta html ... 36
3.2.3 Etiqueta head ... 36
3.2.4 Etiqueta body ... 37
3.2.5 Comentarios ... 37
3.3 CABECERAS Y TÍTULOS ... 37
3.3.1 Elemento title ... 37
3.3.2 Elementos h1..h6 ... 38
3.3.3 Elemento header ... 38
3.3.4 Elemento hgroup ... 39
3.4 INSERCIÓN DE TEXTOS ... 39
3.4.1 Elemento abbr ... 39
3.4.2 Elemento address ... 40
3.4.3 Elemento bdo ... 40
3.4.4 Elementos blockquote y cite ... 41
3.4.5 Elemento code ... 41
3.4.6 Elemento data ... 41
3.4.7 Elemento dfn ... 42
3.4.8 Elemento em ... 42
3.4.9 Elemento i ... 42
3.4.10 Elementos ins y del ... 43
3.4.11 Elemento kbd ... 43
3.4.12 Elemento mark ... 44
3.4.13 Elemento math ... 44
3.4.14 Elemento pre ... 48
3.4.15 Elementos sub y sup ... 48
3.4.16 Elemento var ... 49
3.5 SEPARACIÓN DE BLOQUES DE TEXTOS ... 49
3.5.1 Elemento article ... 49
3.5.2 Elemento aside ... 50
3.5.3 Elemento div ... 50
3.5.4 Elemento footer ... 51
3.5.5 Elemento main ... 52
3.5.6 Elemento nav ... 52
3.5.7 Elemento section ... 53
3.6 INSERCIÓN DE GRÁFICOS ... 53
3.6.1 Elementos figure y figcaption ... 54
3.6.2 Elemento img ... 54
3.6.3 Elemento picture ... 55
3.6.4 Elemento source ... 55
3.7 CREACIÓN DE HIPERVÍNCULOS A OTRAS PÁGINAS ... 56
3.7.1 Elemento a ... 56

3.8 CREACIÓN DE TABLAS 57
3.8.1 Elementos disponibles en HTML5 57
3.9 INSERCIÓN DE DISTINTOS FRAMES EN UNA PÁGINA 60

CAPÍTULO 4. ANIMACIÓN DE GRÁFICOS 65
4.1 CREACIÓN DE ANIMACIONES EN FORMATO VÍDEO 65
4.1.1 Inserción de vídeos en una página web 66
4.2 TRANSFORMACIONES Y EFECTOS 68
4.2.1 Función de escalado (scale) 69
4.2.2 Función de rotación (rotate) 69
4.2.3 Función de sesgado (skew) 70
4.2.4 Función de traslado (translate) 71
4.2.5 Filtros o efectos en CSS 72
4.3 TRANSICIONES 80
4.3.1 Propiedad transition-delay 81
4.3.2 Propiedad transition-duration 82
4.3.3 Propiedad transition-property 82
4.3.4 Propiedad transition-timing-function 83
4.4 ANIMACIONES 84
4.4.1 Propiedad animation-delay 84
4.4.2 Propiedad animation-direction 85
4.4.3 Propiedad animation-duration 85
4.4.4 Propiedad animation-fill-mode 86
4.4.5 Propiedad animation-iteration-count 87
4.4.6 Propiedad animation-name 87
4.4.7 Propiedad animation-play-state 88
4.4.8 Propiedad animation-timing-function 88
4.4.9 Regla keyframes 90
4.4.10 Ejemplos resueltos 91
4.5 EFECTOS CON TEXTOS 94
4.5.1 Efectos estéticos y con movimiento 94
4.5.2 Creación de textos mejorados y con movimiento 95
4.5.3 Adecuación de los efectos a la página web 102
4.6 CREACIÓN DE MAPAS 104
4.6.1 Identificar la ubicación del archivo 104
4.6.2 Incrustación de etiqueta IMG 104
4.6.3 Agregado del atributo usemap 104
4.6.4 Crear el archivo de mapa de imagen 104
4.6.5 Definir las áreas interactivas 105
4.6.6 Guardar y subir el archivo 105
4.7 COMO ARRASTRAR Y COLOCAR IMÁGENES 105
4.7.1 Crear el marcado HTML 105
4.7.2 Estilizar con CSS 106
4.7.3 Agregar el script de JavaScript 106

CAPÍTULO 5. OPTIMIZACIÓN Y LIMITACIONES DEL DISEÑO 109
5.1 REPARTICIÓN DE ESPACIOS DENTRO DE UNA PÁGINA 109
5.1.1 Diseño basado en tablas..... 109
5.1.2 Diseño basado en cajas flexibles (Flexbox) 117
5.1.3 Diseño basado en cuadrículas (Grid Layout)..... 127
5.2 INSERCIÓN DE UN BACKGROUND 133
5.2.1 Propiedades CSS disponibles 133
5.2.2 Adaptación receptiva y adaptativa..... 136
5.3 CREACIÓN DE HIPERVÍNCULOS EN REGIONES DE LA PANTALLA 137
5.3.1 Texto con hipervínculos 138
5.3.2 Imágenes con hipervínculos 138
5.3.3 Mapas de imagen..... 138
5.3.4 CSS para estilizar hipervínculos..... 139
5.4 LIMITACIONES DEL TAMAÑO DE LAS PÁGINAS 139
5.5 OPTIMIZACIÓN DEL TAMAÑO DE LOS GRÁFICOS PARA UNA MAYOR RAPIDEZ 140
5.6 LIMITACIONES DE LA POSICIÓN DE LOS ELEMENTOS..... 141
5.7 FORMATOS DE GRÁFICOS ADMITIDOS 143

REFERENCIAS..... 145

MATERIAL ADICIONAL 147

SOBRE EL AUTOR

Ingeniero informático. UX Manager, Full Stack Analyst Developer y formador certificado por el SEPE, con más de 25 años de experiencia en el campo del desarrollo de sistemas y más de 15 en el campo del diseño de aplicaciones y webs, UX, usabilidad web y accesibilidad web.

Especializado en proveer soluciones integrales de bajo coste y consultorías de Usabilidad Web, Accesibilidad Web y Experiencia de Usuario (UX), además de ofrecer asesoramiento en SEO, optimización de sistemas y páginas web.

Experto en JavaScript, Web Components, JSON, JSON-LD, AJAX, HTML, CSS, MySQL, Oracle, PHP, JSP, GIT, Apache, SSI, LMS, CMS, Prestashop, Photoshop, Canva, Cordova.

- **LinkedIn:** www.linkedin.com/in/pefernandezcasado
- **Sitio web:** https://islavisual.com/

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 puedan 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 automáticas 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 par 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

INTRODUCCIÓN A INTERNET

1.1 INTRODUCCIÓN A INTERNET

Internet se define como una red global de computadoras interconectadas que se comunican entre sí a través de una serie de protocolos estándar. Fue concebida como un sistema descentralizado de comunicaciones que permitía el intercambio de información entre computadoras ubicadas en diferentes partes del mundo y su creación y desarrollo han sido un proceso evolutivo que ha involucrado a múltiples investigadores, instituciones y organizaciones a lo largo de varias décadas.

Según se cree, el origen de Internet se remonta a los primeros experimentos de redes de computadoras en la década de 1960, como el proyecto ARPANET financiado por el Departamento de Defensa de los Estados Unidos. ARPANET fue una red pionera que utilizó el conmutado de paquetes como método de transmisión de datos, sentando las bases para el desarrollo de la red global que conocemos hoy.

No obstante, a medida que la tecnología de redes fue evolucionando, surgieron nuevos protocolos y estándares que permitieron la conexión de redes más amplias y la creación de servicios y aplicaciones innovadoras. Entre los protocolos fundamentales que sustentan Internet podemos encontrar el Protocolo de Internet (IP), que permite la identificación y el enrutamiento de paquetes de datos, y el Protocolo de Control de Transmisión (TCP), que facilita la transmisión confiable de datos entre sistemas finales.

El objetivo principal de Internet es facilitar la comunicación y el intercambio de información a escala mundial y, entre sus principales usos y funciones destaca:

- **Comunicación**:

  Puesto que permite la comunicación instantánea entre personas ubicadas en diferentes partes del mundo a través de servicios como el correo electrónico, la mensajería instantánea, las redes sociales y las videoconferencias.

- **Acceso a la información**:

  Puesto que proporciona acceso a una vasta cantidad de información y recursos en línea, incluidos sitios web, enciclopedias en línea, bibliotecas digitales, blogs y foros de discusión.

- **Comercio electrónico**:

  Puesto que ha transformado la forma en que se realizan las transacciones comerciales, permitiendo la compra y venta de bienes y servicios a través de tiendas online, plataformas de comercio electrónico y sistemas de pago digitales.

- **Educación y entretenimiento**:

  También es importante destacar este punto puesto que se ha convertido en una herramienta invaluable para la educación, ofreciendo recursos educativos en línea, cursos en línea, tutoriales, clases virtuales y plataformas de aprendizaje colaborativo. Y, como no, también se ha vuelto una fuente importante en el entretenimiento ya que ofrece múltiples opciones, como la reproducción de música y vídeo, juegos online, interacción en redes sociales, creación y reproducción de podcasts y publicación de blogs especializados y de contenido multimedia.

1.2 NAVEGADORES DE INTERNET

Los navegadores de Internet son aplicaciones de software diseñadas para permitir a los usuarios acceder y visualizar contenido en Internet. Estas aplicaciones interpretan el código HTML, CSS y JavaScript de los sitios web y presentan el contenido de manera gráfica en la pantalla del usuario. Fueron creados para facilitar la navegación por la World Wide Web, permitiendo a los usuarios buscar, acceder y consumir información en línea de manera intuitiva y eficiente.

El primer navegador web fue desarrollado por Tim Berners-Lee en 1990 y fue denominado como WorldWideWeb, aunque más tarde fue rebautizado como Nexus. Este navegador fue diseñado como parte del proyecto World Wide Web (WWW) que

Berners-Lee había iniciado en el CERN (Organización Europea para la Investigación Nuclear) y su objetivo principal era el de crear un sistema de información distribuido que permitiera a los científicos compartir y acceder a documentos y recursos de investigación de una manera fácil y eficiente.

El navegador WorldWideWeb (o Nexus) fue una aplicación de software bastante rudimentaria en comparación con los navegadores actuales. Sin embargo, sentó las bases para el desarrollo de navegadores más avanzados que surgirían en los años venideros.

En la actualidad, los navegadores web permiten a los usuarios acceder y consumir contenidos de manera rápida, segura y eficiente. Algunas de sus funciones principales son:

- **Navegación por páginas web**:

  Permitiendo a los usuarios visitar sitios web escribiendo la dirección o URL en la barra de direcciones o haciendo clic en enlaces.

- **Renderización de contenido**:

  Interpretando el código HTML, CSS y JavaScript de los sitios web y presentar el contenido de manera adecuada y legible en los dispositivos de usuario, generalmente pantallas.

- **Gestión de sitios favoritos**:

  Permitiendo guardar y organizar sus direcciones o URLs favoritas (actualmente más conocidos como marcadores) para acceder rápidamente a sitios web específicos.

- **Gestión de pestañas**:

  Permitiendo tener abiertas múltiples páginas o sitios web en pestañas separadas dentro de la misma ventana del navegador.

- **Búsqueda Web**:

  Ofreciendo herramientas de búsqueda integradas que permitan a los usuarios buscar información en Internet utilizando motores de búsqueda tan populares como Google, Bing o Yahoo.

- **Seguridad y Privacidad**:

  Proporcionando características de seguridad y privacidad, como bloqueo de ventanas emergentes, navegación privada, protección contra malware y phishing, y administración de cookies.

1.3 EL CORREO ELECTRÓNICO EN INTERNET

El correo electrónico, comúnmente conocido como email, es uno de los servicios más antiguos y fundamentales de Internet. Permite el intercambio de mensajes y archivos digitales entre usuarios a través de la red.

El correo electrónico es una herramienta fundamental en el mundo moderno, ya que proporciona un medio eficiente, rápido y económico para la comunicación y la transferencia de información online. Es ampliamente utilizado tanto a nivel personal como profesional y sigue siendo uno de los servicios más utilizados en Internet, a pesar del surgimiento de nuevas formas de comunicación digital.

Su historia se remonta a la década de 1960, cuando se desarrollaron los primeros sistemas de mensajería electrónica en las redes de computadoras. Sin embargo, el correo electrónico moderno, tal y como hoy lo conocemos, surgió en la década de 1970 con la creación del protocolo SMTP (Simple Mail Transfer Protocol), que es el responsable de estandarizar o normalizar el intercambio de mensajes entre servidores de correo.

Su funcionamiento es, a grandes rasgos, sencillo. Esto es, se realiza a través del intercambio de mensajes entre diferentes direcciones de correo electrónico. Cada dirección de correo electrónico está asociada a una cuenta de correo que puede ser gestionada a través de un cliente de correo electrónico o una interfaz web. Cuando se envía un mensaje, este se transfiere desde el cliente de correo del remitente al servidor de correo saliente (SMTP) que lo envía al servidor de correo entrante (POP3, IMAP) del destinatario, desde donde se entrega al cliente de correo del destinatario.

Pero, ¿qué se necesita para que se produzca esta comunicación? Pues, básicamente, tres cosas:

- **Dirección de correo electrónico**:

  Habitualmente definida como una cadena de caracteres que identifica de manera única una cuenta de correo electrónico y que se compone de un nombre de usuario seguido del símbolo “@” y un nombre de dominio de proveedor.

- **Cliente de correo electrónico**:

  Una aplicación informática utilizada para enviar, recibir, organizar y gestionar todos los posibles correos electrónicos. Entre los más populares podríamos destacar Gmail, Outlook y Thunderbird.

- **Servidor de correo**:

  Un sistema que almacene, envíe y reciba los mensajes de correo electrónico en un disco duro o sistema de almacenamiento interno o externo. Los servidores de correo saliente (SMTP) se encargan de enviar mensajes, mientras que los servidores de correo entrante (POP3, IMAP) almacenan y entregan los mensajes entrantes a los destinatarios.

Por último, cabe destacar que el correo electrónico se utiliza para una gran variedad de objetivos o propósitos, tanto personales como profesionales, que incluyen cosas como comunicación personal y social, el intercambio de correo empresarial y comunicación interna, el envío de documentos, archivos y otros contenidos multimedia, acceso a boletines informativos y comunicaciones comerciales y confirmaciones de registro, recibos y notificaciones automáticas.

1.4 OTRAS APLICACIONES DE INTERNET

Además de las funciones básicas de comunicación, como es caso del correo electrónico o el acceso a la información, Internet puede proveer de otras aplicaciones y usos como son:

- **Telemedicina**:

  Permitiendo la prestación de servicios médicos a distancia, como consultas médicas en línea, telecirugía, telediagnóstico, seguimiento de pacientes y educación médica a distancia.

- **Trabajo remoto**:

  Facilitando la colaboración en equipo, la comunicación entre colegas, la gestión de proyectos y la realización de reuniones virtuales a través de herramientas como videoconferencias y aplicaciones de productividad online.

- **IoT (Internet de las Cosas)**:

  Posibilitando la conexión de dispositivos inteligentes y sensores a la red, permitiendo la recopilación y el intercambio de datos en tiempo real para una variedad de aplicaciones, como domótica, salud, agricultura, industria y transporte.

- **Banca online**:

  Permitiendo a los usuarios realizar transacciones bancarias, pagos de facturas, transferencias de fondos, consultas de saldo y gestión de cuentas desde cualquier lugar con acceso a Internet.

- **Gobierno electrónico**:

  Ofreciendo servicios online, como trámites administrativos, pago de impuestos, solicitud de documentos y acceso a información pública, a través de portales gubernamentales y plataformas electrónicas.

- **Redes sociales y aplicaciones de mensajería instantánea**:

  Permitiendo a los usuarios conectarse, compartir contenido, interactuar y comunicarse entre sí. Estas plataformas se utilizan para diversos fines, como mantenerse en contacto con amigos y familiares, establecer contactos profesionales, compartir noticias y promocionar productos o servicios.

- **E-learning**:

  Transformando la educación con la proliferación de plataformas de aprendizaje online, cursos virtuales, tutoriales, recursos educativos abiertos y herramientas de colaboración que permiten el aprendizaje a distancia y la formación continua.

- **Streaming de contenido**:

  Creando y dando acceso a servicios de streaming de música, películas, series de televisión, podcasts y vídeos online, los cuales permiten a los usuarios acceder a una amplia variedad de contenido multimedia bajo demanda a través de Internet.

- **Comunicación VoIP**:

  Permitiendo realizar llamadas de voz y vídeo a través de Internet, a menudo de forma gratuita o a costos reducidos en comparación con las llamadas tradicionales.

En realidad, éstas son sólo algunas de las muchas aplicaciones que posee Internet en la actualidad. Esto es porque la versatilidad y la omnipresencia de Internet ha transformado fundamentalmente la forma en que vivimos, trabajamos, nos comunicamos y nos relacionamos con todo lo que nos rodea.

1.5 HERRAMIENTAS DE DISEÑO DE PÁGINAS WEB

Existen diversas herramientas de diseño de páginas web disponibles en el mercado, cada una con sus propias características y funcionalidades. A continuación, se presentan algunas de las herramientas más populares para diseñar y desarrollar páginas web:

- **Adobe Dreamweaver**

  Dreamweaver es una herramienta de desarrollo web líder en la industria que proporciona un entorno visual y de código para diseñar, codificar y administrar sitios web. Ofrece funciones de diseño responsivo, previsualización en tiempo real y compatibilidad con numerosos lenguajes de programación.

- **WordPress**:

  WordPress es un sistema de gestión de contenidos (CMS) que permite crear sitios web dinámicos y flexibles mediante la utilización de temas y plugins. Es especialmente popular para la creación de blogs y sitios web de contenido, ofreciendo una interfaz intuitiva y una amplia comunidad de desarrolladores.

- **Wix**:

  Wix es una plataforma de diseño y desarrollo web online que proporciona herramientas drag-and-drop para crear sitios web de aspecto profesional sin necesidad de conocimientos de programación. Ofrece una amplia variedad de plantillas y widgets personalizables para adaptarse a las necesidades de cualquier proyecto.

- **Squarespace**:

  Squarespace es otra plataforma de diseño web que ofrece plantillas elegantes y personalizables, así como herramientas integradas para la gestión de contenido, comercio electrónico y análisis de sitios web. Es ideal para diseñadores y emprendedores que buscan crear sitios web visualmente impactantes.

- **Notion**:

  Notion es una plataforma de productividad y gestión de proyectos “no code” que permite a los usuarios crear, organizar y colaborar en una variedad de contenido, como notas, documentos, listas de tareas, bases de datos, calendarios y mucho más. Se destaca por su flexibilidad y capacidad

para adaptarse a una amplia gama de casos de uso, desde la organización personal hasta la colaboración en equipo en entornos empresariales.

- **Sketch**:

  Sketch es una herramienta de diseño gráfico centrada en la creación de interfaces de usuario (UI) y diseño de experiencia de usuario (UX). Es ampliamente utilizado por diseñadores para crear wireframes, mockups y diseños de sitios web y aplicaciones móviles.

- **Adobe XD**:

  Adobe XD es otra herramienta de diseño de experiencia de usuario (UX) y prototipado que permite a los diseñadores crear diseños interactivos, wireframes y prototipos de alta fidelidad para sitios web y aplicaciones móviles.

- **Bootstrap, Foundation, Bulma**:

  Bootstrap, Foundation, Bulma y otros so frameworks de desarrollo front-end que proporcionan un conjunto de herramientas y componentes preestablecidos para crear interfaces web responsivas y adaptables. En este contexto, quizás el más popular sea Bootstrap, por su facilidad integración con otros frameworks y sistemas, por su facilidad de uso y su enfoque en el diseño receptivo.

2

PLANIFICACIÓN DE LA WEB

2.1 DISEÑO DEL CONTENIDO

El diseño de contenido web se refiere al proceso de planificación, creación y organización de todos los elementos visuales, auditivos y escritos que componen un sitio web. Esto incluye textos, imágenes, vídeos, gráficos, animaciones y otros recursos multimedia que se utilizan para comunicar mensajes y proporcionar una experiencia atractiva y efectiva a los usuarios en línea.

El diseño de contenido web tiene como objetivo principal crear una estructura visualmente atractiva y funcional que guíe a los usuarios a través del sitio web de manera intuitiva y efectiva proporcionándoles la información que estén buscando de manera clara y concisa. Por tanto, si queremos conseguir un buen diseño de contenido, deberemos tener en cuenta cosas como la usabilidad, la accesibilidad y la experiencia del usuario.

Algunos aspectos importantes del diseño de contenido web incluyen:

- **Arquitectura de la información**:
  Esto implica la elección de una estructura y organización del contenido, que incluye desde la creación de categorías, subcategorías y menús de navegación, hasta la jerarquía de la información para facilitar la búsqueda y la comprensión.

- **Diseño visual**:

  Esto implica la selección de colores, tipografías, imágenes y otros elementos visuales que se utilizarán en el sitio web para transmitir la marca y el mensaje de manera efectiva y atractiva.

- **Optimización para dispositivos**:

  Esto implica asegurarse de que el diseño sea compatible y se vea correctamente en todos los dispositivos y tamaños de pantalla, incluyendo computadoras de escritorio, tabletas y dispositivos móviles.

- **Posicionamiento SEO**:

  Esto implica la optimización del contenido para motores de búsqueda (SEO) como Google o Yahoo a través de palabras clave relevantes, metadatos y otros elementos para mejorar la visibilidad del sitio web en los resultados de búsqueda.

- **Usabilidad y accesibilidad**:

  Esto implica garantizar que el contenido sea fácil de usar y accesible para todos los usuarios, incluidos aquellos que poseen discapacidad parcial o total visual, auditiva, intelectual, cognitiva, motriz o sea del tipo que sea.

2.2 PROPÓSITO Y AUDIENCIA DE LA WEB

El propósito y la audiencia de un sitio web son aspectos interrelacionados que influyen en su diseño y desarrollo. Definir claramente estos elementos desde el principio es fundamental para crear un sitio web eficiente, efectivo y exitoso que cumpla con los objetivos de la empresa u organización y satisfaga las necesidades de los usuarios.

2.2.1 Propósito de la web

El propósito de un sitio web se refiere a la razón principal por la cual fue creado y qué objetivos busca cumplir. Esto incluye cosas como:

- Promover y vender productos o servicios.
- Proporcionar información sobre una empresa, organización o causa.
- Educar y entretener a los visitantes.
- Facilitar la comunicación y la interacción entre usuarios.

- Generar leads y convertir visitantes en clientes potenciales.
- Brindar soporte y atención al cliente.
- Establecer una presencia en línea y construir una marca.

Por todo esto, y mucho más, es importante definir claramente el propósito del sitio web desde el principio, ya que nos ayudará a tomar todas las decisiones de diseño, contenido y funcionalidad.

2.2.2 Audiencia de la web

La audiencia de un sitio web se refiere al grupo de personas que se espera que visiten y utilicen el sitio. Si comprendemos y tomamos una buena idea de quiénes son nuestros usuarios, sus necesidades, intereses y comportamientos, habremos adquirido una buena información sobre nuestra audiencia. Para ello, podremos valernos de indicadores o aspectos como:

- **Demografía**:

  La edad, género, ubicación geográfica, nivel socioeconómico, etc.

- **Intereses y preferencias**:

  Aspectos como, ¿qué temas, productos o servicios les interesan y si tienen alguna tendencia predefinida?

- **Comportamiento online**:

  Aspectos como, ¿cómo suelen interactuar online y qué dispositivos utilizan?

- **Necesidades y objetivos**:

  Aspectos como, ¿qué están buscando cuando visitan el sitio y qué acciones esperan realizar?

Por tanto, identificar y comprender a la audiencia objetivo es crucial para diseñar un sitio web que satisfaga sus necesidades y expectativas. Esto implica crear contenido relevante, ofrecer una experiencia de usuario intuitiva y proporcionar funcionalidades que resuelvan sus problemas y les brinden valor.

2.3 DISEÑO DE LA APARIENCIA

El diseño de la apariencia web es un aspecto fundamental del diseño web que implica la creación o invención de la estética de un sitio web para mejorar la experiencia del usuario y reflejar la identidad de marca de la empresa. En este proceso, no sólo será importante el tema visual, sino que, además, será crucial tener en cuenta aspectos como la usabilidad, la accesibilidad y la experiencia de usuario

Entre los elementos más importantes se puede destacar:

- **Diseño visual**:

  Aquí se incluye la selección de colores, tipografías, imágenes, iconos y otros elementos visuales que se utilizarán en el sitio web. Todo ello, deberá ser coherente y consistente con la identidad de marca para ayudar a transmitir su mensaje y sus valores.

- **Diseño de la interfaz de usuario (UI)**:

  La interfaz de usuario se refiere a la forma en que los elementos visuales se organizan y presentan en las páginas para facilitar la interacción del usuario. Esto incluye desde los menús de navegación y botones, hasta los formularios y barras laterales, entre otros elementos de la interfaz.

- **Usabilidad**:

  El diseño de la apariencia web debe optimizarse para mejorar el entendimiento, comprensión, legibilidad y rendimiento, entre otros factores. Por tanto, esto implicará asegurarse de que la navegación sea intuitiva, los elementos interactivos sean fáciles de entender y utilizar, y el contenido sea fácilmente accesible para los usuarios.

- **Diseño receptivo**:

  Es importante que el diseño sea responsive, es decir, que sea fluido y adaptable para que se vea correctamente en todos los dispositivos y tamaños de pantalla, incluyendo computadoras de escritorio, tabletas y teléfonos móviles.

- **Accesibilidad**:

  Aunque la usabilidad ya ofrece una buena respuesta ante la experiencia de usuario, también será recomendable tener en cuenta la accesibilidad para garantizar que el sitio sea usable por personas con discapacidad parcial o total, sean de la índole que sean. Esto incluirá, entre otras cosas, la

inclusión de textos alternativos para imágenes, la navegación por teclado y contrastes adecuados entre el texto y el fondo. No olvidemos que "si una aplicación o página es usable, puede que no sea accesible, pero, si es accesible, seguro que es usable".

- **Estética**:

  Es evidente que, además de ser funcional, usable y accesible, el diseño también deberá ser estéticamente agradable y atractivo para los usuarios. Esto implicará la aplicación de un diseño minimalista, limpio y moderno, la atención al detalle y una buena coherencia o consistencia en todas las páginas del sitio.

2.4 CONSTRUIR EL SITIO

Construir un sitio web implica seguir unos pasos y tomar una serie de decisiones para asegurar que el sitio cumpla con los objetivos, siendo efectivo para todos los usuarios. A continuación, se muestra los pasos clave o básicos para construir un sitio web de forma correcta:

- **Definir el propósito y los objetivos del sitio web**:

  Antes de comenzar a construir el sitio, es importante tener claro cuál será su propósito principal y qué objetivos se esperarán lograr con él. Esto ayudará a tomar todas las decisiones de diseño y desarrollo que se requieran.

- **Elegir una plataforma o herramienta de desarrollo**:

  Una vez definidos el propósito y los objetivos del sitio, será una buena idea seleccionar o elegir cómo se va a desarrollar el sitio. Existen muchas opciones disponibles para construir un sitio web que van desde plataformas de gestión de contenido (CMS) como WordPress, hasta constructores de sitios web "no code" basados en metodologías de arrastrar y soltar como Wix o Squarespace. No obstante, lo más frecuente y profesional es recurrir a la codificación manual utilizando lenguajes como HTML, CSS y JavaScript.

- **Registrar un nombre de dominio y elegir un proveedor de alojamiento**:

  Se trata de elegir un nombre de dominio adecuado para la marca y el sitio web. Es decir, una dirección web única que sea fácilmente identificable y recordable como pueda ser https://www.amazon.com. El registro se

podrá realizar a través de un registrador de dominios el cual, a su vez, es posible que también nos proporcione un servicio de alojamiento web confiable para alojar su sitio.

- **Diseñar la estructura y el diseño del sitio**:

  Una vez tenemos los requisitos y tecnologías a utilizar, será conveniente, si no obligatorio, diseñar la estructura y el diseño del sitio web en papel o mediante herramientas de diseño como Sketch. Esto incluirá, o deberá incluir, la planificación de la arquitectura de la información, la navegación, el diseño visual y la disposición de los elementos en la página.

- **Desarrollar y construir el sitio**:

  Cuando se tenga la estructura y diseño del sitio, será el momento de trabajar con la plataforma o herramienta elegida. Esto podrá implicar tener unos u otros conocimientos, instalar plantillas y/o temas específicos, personalizar de diseño y funcionalidades, crear diferentes tipos de página y establecer qué y cómo se realizará la carga de los contenidos.

- **Optimizar para posicionamiento SEO**:

  A medida que construyen los sitios, hay que asegurarse de optimizarlos para los motores de búsqueda (SEO). Esto, generalmente, incluirá el uso de palabras clave relevantes, metadatos, microdatos a través de alguna anotación como JSON-LD y la creación de contenidos legibles y de calidad.

- **Pruebas y revisión**:

  Una vez que el sitio esté "terminado", se deberán realizar pruebas exhaustivas para asegurarse de que todo funcione adecuadamente en todos los navegadores y dispositivos, y de que cumpla con sus objetivos y requisitos.

- **Lanzamiento y promoción**:

  Cuando se esté satisfecho con los test y resultados que nos han proporcionado las pruebas, será el momento de publicarlo y comenzar a promocionarlo para atraer nuevos usuarios y visitas. Esto podrá implicar el uso de redes sociales, el envío de correos electrónicos y/o utilizar estrategias de marketing digital, entre otros métodos.

- **Mantenimiento y actualización**:

  Finalmente, se deberá realizar un mantenimiento regular que podrá incluir la aplicación de parches de seguridad y la actualización de contenidos según sea necesario.

2.5 ANALIZAR LA ESTRUCTURA

Analizar la estructura de un sitio web es importante porque dependiendo de cómo se haga, los resultados pueden afectar a la organización y navegación a nivel global o general. A continuación, se comentan algunos elementos clave a tener en cuenta para conseguir unos buenos resultados:

- **Mapa web**:

  Una de las cosas más interesantes que resultan del análisis de la estructura es el mapa del sitio. Esto es, una representación visual de la estructura y organización de todo el sitio web que muestra todas las páginas y secciones y cómo se relacionan entre sí los diferentes elementos.

- **Navegación**:

  La navegación del sitio web será fundamental para la experiencia del usuario. Analizar cómo se presentan las opciones de navegación principal, los enlaces de navegación secundarios y cualquier otra forma de acceso a las diferentes secciones del sitio será una buena forma de analizar la estructura de un sitio para construir, entre otras cosas, su mapa web.

- **Arquitectura de la información**:

  En este contexto es donde se ve la organización y estructura de los contenidos. Esto nos mostrará cómo se agrupan y categorizan las páginas y secciones, y cómo se facilita la navegación entre ellas.

- **URLs y enlaces internos**:

  Cuando se analiza la estructura de una web, es habitual encontrar varias URLs entre las diferentes páginas. En este sentido, las URLs, internas o no, deben ser claras y descriptivas para facilitar la navegación dentro del sitio.

- **Jerarquía de contenido**:

  Se trata de analizar la jerarquía del contenido en el sitio, incluida la relación entre las páginas principales y las páginas secundarias, así como la profundidad de la estructura de navegación.

- **Etiquetas y metadatos**:

  Se trata de revisar las etiquetas y metadatos utilizados en el sitio, como los títulos de página, descripciones meta y etiquetas de encabezado. Estos elementos juegan un papel importante en la optimización para motores de búsqueda y la comprensión del contenido para los usuarios y herramientas de asistencia como los lectores de pantalla, lupas, etc.

- **Compatibilidad con dispositivos**:

  Es importante analizar cómo se adapta la estructura del sitio a diferentes dispositivos y tamaños de pantalla. Por tanto, la estructura del sitio deberá ser receptiva y ofrecer una experiencia de usuario consistente en todos los dispositivos.

- **Accesibilidad**:

  En este último punto, se trata de revisar la estructura del sitio para garantizar que sea accesible para todos los usuarios, incluidos aquellos con discapacidad parcial o total. Esto podrá implicar el uso de etiquetas semánticas, la navegación por teclado y otras prácticas de accesibilidad como la aplicación de la WAI-ARIA.

2.6 ORGANIZAR ARCHIVOS

Organizar los archivos de una forma adecuada es fundamental para mantener una estructura limpia y clara que facilite su mantenimiento y mejore la eficiencia durante el desarrollo. A continuación, se muestran o comentan algunas prácticas comunes para organizar los archivos de un sitio web:

- **Crear una estructura de carpetas lógica**:

  Organizar los archivos en carpetas de forma que reflejen la estructura y la navegación del sitio web de manera lógica y efectiva. Por ejemplo, es frecuente tener carpetas separadas para las páginas principales, los archivos de estilo (CSS), los scripts (JavaScript), las imágenes y demás recursos.

- **Separar el contenido estático y dinámico**:

  Si nuestro sitio web utiliza un sistema de gestión de contenido (CMS) o genera contenido dinámicamente, suele ser conveniente separar los archivos estáticos (como HTML, CSS y JavaScript) de los archivos dinámicos (como plantillas de CMS, scripts de servidor y bases de datos).

- **Utilizar una nomenclatura consistente**:

  Es importante ser coherente a la hora de nombrar los archivos y carpetas para facilitar la identificación y el mantenimiento. Por ejemplo, use nombres descriptivos y significativos que reflejen el contenido o la función de cada archivo como “css”, “images” o “scripts”.

- **Agrupar archivos relacionados**:

  Agrupar los archivos relacionados en la misma carpeta suele ser una buena idea para facilitar el acceso y gestión. Por ejemplo, es habitual mantener todos los archivos relacionados con el diseño en una carpeta de estilos en el directorio “css” y todos los archivos relacionados con scripts en la carpeta “scripts”.

- **Evitar la sobrecarga de carpetas**:

  Es importante no sobrecargar las carpetas con demasiados archivos o subcarpetas. Si una carpeta contiene demasiados archivos, lo más conveniente será dividirla en subcarpetas más pequeñas para mejorar la organización.

Además de estas directrices, también podrá sernos de utilidad:

- **Utilizar un sistema de control de versiones**:

  Si es posible, siempre será una buena idea utilizar un sistema de control de versiones como Git para gestionar y controlar los cambios en los distintos archivos que componen el sitio web. Esto nos facilitará la vida bastante, mejorará la colaboración entre desarrolladores y permitirá obtener un historial de cambios.

- **Realizar copias de seguridad regularmente**:

  Independientemente de cómo se organicen los archivos, debemos asegurarnos de realizar copias de seguridad regulares de todo el sitio para estar protegidos ante la pérdida de datos y garantizar la continuidad del negocio.

3

LENGUAJE HTML

3.1 PROGRAMACIÓN EN HTML

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

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

Ilustración 2.2. Etiquetado de un campo de formulario en HTML

3.1.1.1 DTDS DE HTML

HTML5

```
<!DOCTYPE html>
```

3.1.1.2 DTDS DE MATHML

MathML 2.0

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

3.1.1.3 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”>
```

3.1.1.4 DTDS DE XHTML

XHTML1.1

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

3.2.2 Etiqueta html

```
<html lang=”es”>...</html>
```

3.2.3 Etiqueta head

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

3.2.4 Etiqueta body

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

3.2.5 Comentarios

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

3.3 CABECERAS Y TÍTULOS

3.3.1 Elemento title

```
<title>Curso de creación de páginas web</title>
```

3.3.2 Elementos h1..h6

```
<h1>Esto es un encabezado de nivel 1</h1>
<h2>Esto es un encabezado de nivel 2</h2>
<h3>Esto es un encabezado de nivel 3</h3>
<h4>Esto es un encabezado de nivel 4</h4>
<h5>Esto es un encabezado de nivel 5</h5>
<h6>Esto es un encabezado de nivel 6</h6>
```

3.3.3 Elemento header

```
<article>
    <header>
        <h2>HTML5</h2>
    </header>
    ...
</article>
```

3.3.4 Elemento hgroup

```
<hgroup>
    <h1>Fast and Furious 4</h1>
    <h2>Aún más rápido</h2>
</hgroup>
```

3.4 INSERCIÓN DE TEXTOS

3.4.1 Elemento abbr

```
<abbr title=”Cascading Style Sheets”>CSS</abbr> es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.
```

NOTA

Al poner el puntero del ratón encima de CSS debería aprecer un tooltip con el mensaje “Cascading Style Sheets”.

3.4.2 Elemento address

```
<address>
    Escrito por Pablo Enrique Fernández Casado.
    Visita <a href=”https://ejemplo.com”>Ejemplo.com</a>
    Castellana 58, local
    28046 Madrid
    España
</address>
```

```
<address>
    Email de contacto:
    <a href=”mailto:ejemplo@gmail.com”>ejemplo@gmail.com</a><br>
    Teléfono: <a href=”tel:+34999999999”>(+34) 999.999.999</a>
</address>
```

3.4.3 Elemento bdo

```
<p dir=”ltr”>Esta palabara arábica <bdo dir=”rtl”>ARABIC PLACEHOLDER</
bdo>, está escrita de izquierda a derecha, pero se muestra al revés.</
p>
```

3.4.4 Elementos blockquote y cite

```
<blockquote cite=”https://blog.com/einstein”>
    Hay dos cosas infinitas, el Universo y la estupidez humana
</blockquote>
```

```
<p>
    <cite>
        Hay dos cosas infinitas, el Universo y la estupidez humana
    </cite>, dicho por Albert Einstein
</p>
```

3.4.5 Elemento code

```
<code>
    <script type=”text/javascript”>
        document.querySelector(“body”).style.fontSize = “14px”;
    </script>
</code>
```

3.4.6 Elemento data

Un ejemplo podría ser:

```
<ul>
    <li><data value="3967381398">Producto pequeño</data></li>
    <li><data value="3967381399">Producto mediano</data></li>
    <li><data value="3967381400">Producto grande</data></li>
</ul>
```

3.4.7 Elemento dfn

```
<p>
    El <dfn>HTML</dfn> es un lenguaje de marcado para hipertextos.
</p>
```

3.4.8 Elemento em

:

```
<p>
    Este texto no tiene énfasis,
    <em>pero este texto sí está con énfasis</em>
</p>
```

3.4.9 Elemento i

```
<p>
    Este texto no tiene énfasis,
    <i>pero este texto sí está con énfasis</i>
</p>
```

3.4.10 Elementos ins y del

```
<p>
    El cometa <del>C/2020 F3</del> <ins>Neowise</ins>, descubierto ...
</p>
```

3.4.11 Elemento kbd

```
<kbd>alt + S</kbd>
```

```
<p>
    Pulse <kbd><kbd style="border: 1px solid #000; border-radius: 4px;
padding: 2px;">Ctrl</kbd> + <kbd style="border: 1px solid #000; border-
radius: 4px; padding: 2px;">R</kbd></kbd> para recargar la página.
</p>
```

3.4.12 Elemento mark

```
<p>
    Los <mark>elementos P no deben contener etiquetas que no sean de
texto</mark>. Esto es, no es aconsejable introducir en una etiqueta de
párrafo un elemento DIV, SECTION, ARTICLE, ...
</p>
```

Los elementos P no deben contener etiquetas que no sean de texto. Esto es, no es aconsejable introducir en una etiqueta de párrafo un elemento DIV, SECTION, ARTICLE, ...

3.4.13 Elemento math

3.4.13.1 EJEMPLO DE SUPERÍNDICES

```
<math>
    <msup>
        <mi>n</mi>
        <mn>7</mn>
    </msup>
</math>
```

El resultado debiera ser similar a:

$$n^7$$

3.4.13.2 EJEMPLO DE SUBÍNDICES

```
<math>
    <msub>
        <mi>n</mi>
        <mn>7</mn>
    </msub>
</math>
```

El resultado debiera ser similar a:

$$n_7$$

3.4.13.3 EJEMPLO DE FRACCIONES

```
<math>
    <mfrac>
        <mn>1</mn>
        <mn>2</mn>
    </mfrac>
</math>
```

El resultado debiera ser similar a:

$$\frac{1}{2}$$

3.4.13.4 EJEMPLO DE RAÍCES

```
<math>
    <mroot>
        <mn>-8</mn>
        <mn>3</mn>
    </mroot>
</math>
```

El resultado debiera ser similar a:

$$\sqrt[3]{-8}$$

3.4.13.5 EJEMPLO DE SUMATORIOS

```
<math>
    <mrow>
        <munderover>
            <mo>∑</mo>
            <mrow>
                <mi>n</mi>
                <mo>=</mo>
                <mn>1</mn>
            </mrow>
            <mrow>
                <mo>+</mo>
                <mn>∞</mn>
            </mrow>
        </munderover>
        <mfrac>
            <mn>1</mn>
            <msup>
                <mi>n</mi>
                <mn>2</mn>
            </msup>
        </mfrac>
    </mrow>
</math>
```

El resultado debiera ser similar a:

$$\sum_{n=1}^{+\infty} \frac{1}{n^2}$$

3.4.13.6 EJEMPLO DE MATRICES

```
<math>
    <mrow>
        <mo>[</mo>
        <mtable>
            <mtr>
                <mtd> <mn style="color: var(--color2-bg);">x</mn> </
mtd>
                <mtd> <mn>1</mn> </mtd>
            </mtr>
            <mtr>
                <mtd> <mn>2</mn> </mtd>
                <mtd> <mn>3</mn> </mtd>
            </mtr>
        </mtable>
        <mo>]</mo>
    </mrow>
</math>
```

El resultado debiera ser similar a:

$$\begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}$$

3.4.13.7 EJEMPLO DE INTEGRALES

```
<math>
    <munderover>
        <mo>∫</mo>
        <mi>a</mi>
        <mi>b</mi>
    </munderover>
    <mrow>
        <mo>(</mo>
        <mn>5</mn>
        <mi>x</mi>
        <mo>+</mo>
        <mn>2</mn>
        <mi>cos</mi>
```

```
        <mrow>
            <mo>(</mo>
            <mi>x</mi>
            <mo>)</mo>
        </mrow>
        <mo>)</mo>
    </mrow>
    <mi>dx</mi>
</math>
```

El resultado debiera ser similar a:

$$\int_a^b \left(5x + 2\cos\left(x\right)\right)\mathrm{dx}$$

3.4.14 Elemento pre

```
<pre>
    <p>
        Los  espacios repetidos y
        Saltos de línea de    este   elemento se muestran tal cuál!
    </p>
</pre>
```

El resultado debiera ser similar a:

Los espacios repetidos y saltos de línea de este elemento se muestran tal cual.

3.4.15 Elementos sub y sup

Un ejemplo con SUB podría ser:

```
<p>La fórmula del agua es H<sub>2</sub>O</p>
```

```
<p>E = MC<sup>2</sup></p>
```

Resultado

E = MC^2

3.4.16 Elemento var

```
<var>x</var> = Millones de personas;
```

3.5 SEPARACIÓN DE BLOQUES DE TEXTOS

3.5.1 Elemento article

```
<article>
    <header>
        <h2>HTML5</h2>
    </header>

    <div>
```

```
        <p> El lenguaje HTML5 (HyperText Markup Language Versión 5) es
un lenguaje de marcado de hipertexto que está vigente desde el año 2014
y puede ser utilizado para...</p>
    </div>

    <footer>
        <a href=”html-usos.html”>Seguir leyendo</a>
    </footer>
</article>
```

3.5.2 Elemento aside

```
<aside>
    <h3>Artículos relaccionados</h3>
    <ul>
        <li><a href=”#”>Artículo 1</a></li>
        <li><a href=”#”>Artículo 2</a></li>
        <li><a href=”#”>Artículo 3</a></li>
    </ul>
</aside>
```

3.5.3 Elemento div

3.9 INSERCIÓN DE DISTINTOS FRAMES EN UNA PÁGINA

```
<div class=” vídeo-responsive”>
    <iframe src=”https://www.youtube.com/embed/0sAc60jzKv4”
            frameborder=”0”
            allow=”accelerometer; autoplay; encrypted-media;
                   gyroscope; picture-in-picture”
            allowfullscreen>
    </iframe>
</div>
```

```
<style>
    .vídeo-container {
        overflow: hidden;
        position: relative;
        width:100%;
    }

    .vídeo-container::after {
        padding-top: 56.25%;        /* Esto es porque 9 es el 56.25% de 16
        display: block;
        content: ‘’;
    }

    .vídeo-container iframe {
        position: absolute;
        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
    }
</style>

<div class=”vídeo-container”>
    <iframe allow=”accelerometer; autoplay; encrypted-media; gyroscope;
                   picture-in-picture”
            allowfullscreen
            frameborder=”0”
            src=”https://www.youtube.com/embed/x2D7jHfitzk”>
    </iframe>
</div>
```

4

ANIMACIÓN DE GRÁFICOS

4.1 CREACIÓN DE ANIMACIONES EN FORMATO VÍDEO

4.1.1 Inserción de vídeos en una página web

4.1.1.1 ELEMENTO SOURCE

```
<picture>
  <source srcset=”./img/land-laptop.png” media=”(min-width: 1366px)” />
  <source srcset=”./img/land-tablet.png” media=”(min-width: 900px)” />
  <source srcset=”./img/land-mobile.png” media=”(min-width: 768px)” />
</picture>
```

4.1.1.1.1 Atributo SRCSET

4.1.1.2 ELEMENTO VÍDEO

```
<vídeo controls>
    <source src=” el-quinto-elemento.mp4” type=”audio/mp4”>

    El navegador no soporta la etiqueta vídeo.
</vídeo>
```

Entre los atributos que admite en su configuración, hay que destacar la

4.2 TRANSFORMACIONES Y EFECTOS

```
div {
    width: 100px;
    height: 100px;
    background: linear-gradient(0deg, black 0, black 50%, white);
    border: 2px solid #000;
    margin: 0;
    padding: 0;
}
```

4.2.1 Función de escalado (scale)

Ejemplos:

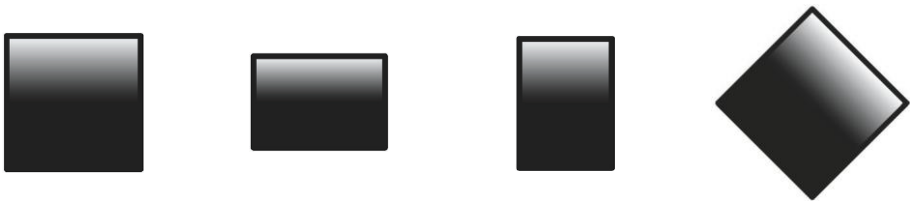
```
div { transform: scale(0.5); }
div { transform: scale(0.5, 1.5); }
```

4.2.2 Función de rotación (rotate)

Ejemplos:

```
div { transform: rotate(45deg); }
div { transform: rotate(-45deg); }
```

Si se desea que el giro se realice en un eje en particular, se debe recurrir a ROTATEX, ROTATEY o ROTATEZ.



Forma original **rotateX(45deg)** **rotateY(45deg)** **rotateZ(45deg)**

4.2.3 Función de sesgado (skew)

Forma original **Inclinación positiva** **Inclinación negativa**

Ejemplos:

```
div { transform: skew(45deg); }
div { transform: skew(-45deg); }
```

4.2.4 Función de traslado (translate)

La función TRANSLATE permite mover un objeto respecto en cualquiera de

Original **Desplazamiento positivo** **Desplazamiento negativo**

Ejemplos:

```
div { transform: translate(100px); }
div { transform: translate(-100px); }
```

Forma original **translateX(10px)** **translateY(10px)** **translateZ(10px)**

Ejemplos:

```
div { transform: perspective(100px) translateZ(10px); }
```

4.2.5 Filtros o efectos en CSS

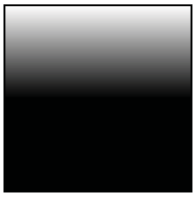


Imagen Original

Desenfoque Gaussiano 2px

Desenfoque Gaussiano 4px

Ejemplos:

```
div { filter: blur(0px); /* Imagen izquierda */ }
div { filter: blur(2px); /* Imagen central */ }
div { filter: blur(4px); /* Imagen derecha */ }
```

4.2.5.1 BRILLO O ILUMINACIÓN (BRIGHTNESS)

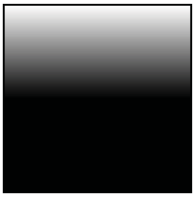
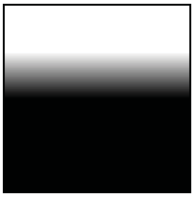


Imagen Original



Iluminación al 200%

Iluminación al 400%

Ejemplos:

```
div { filter: brightness(0); /* Imagen izquierda */ }
div { filter: brightness(2); /* Imagen central */ }
div { filter: brightness(4); /* Imagen derecha */ }
```

4.2.5.2 CONTRASTE (CONTRAST)

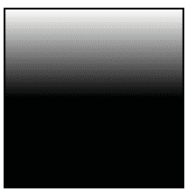


Imagen Original

filter: contrast(0.5)

filter: contrast(0);

Ejemplos:

```
div { filter: contrast(1);   /* Imagen izquierda */ }
div { filter: contrast(0.5); /* Imagen central */ }
div { filter: contrast(0);   /* Imagen derecha */ }
```

4.2.5.3 SOMBRA PARALELA (DROP-SHADOW)

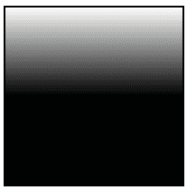
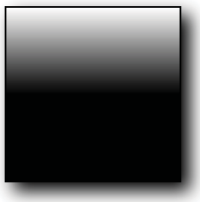


Imagen Original

Sombra sin desenfoque

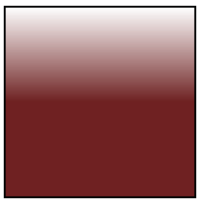


Sombra con desenfoque

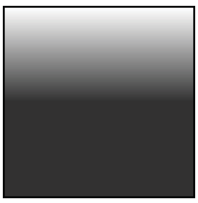
Ejemplos:

```
div { filter: drop-shadow(0px 0px 0px #000);     /* Imagen izquierda */
}
div { filter: drop-shadow(10px 10px 0px #000);   /* Imagen central */ }
div { filter: drop-shadow(10px 10px 10px #000);  /* Imagen derecha */ }
```

Imagen Original



Escala de grises al 50%

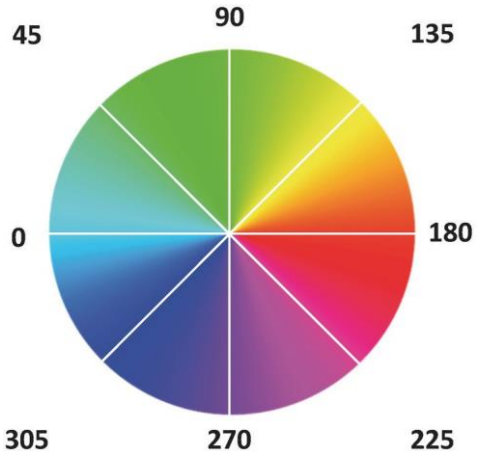


Escala de grises al 100%

Ejemplos:

```
div { filter: grayscale(0);   /* Imagen izquierda */ }
div { filter: grayscale(0.5); /* Imagen central */ }
div { filter: grayscale(1);   /* Imagen derecha */ }
```

4.2.5.4 ROTACIÓN DE COLOR (HUE-ROTATE)



Círculo cromático natural degradado. Fuente: Wikipedia, la enciclopedia libre.

Imagen Original **Rotación de 90 grados** **Rotación de 180 grados**

Ejemplos:

```
div { filter: hue-rotate(0deg);    /* Imagen izquierda */ }
div { filter: hue-rotate(90deg);   /* Imagen central */ }
div { filter: hue-rotate(180deg);  /* Imagen derecha */ }
```

4.2.5.5 INVERSIÓN (INVERT)

Imagen Original | Inversión al 50% | Inversión al 100%

Ejemplos:

```
div { filter: invert(0);   /* Imagen izquierda */ }
div { filter: invert(0.5); /* Imagen central */ }
div { filter: invert(1);   /* Imagen derecha */ }
```

4.2.5.6 OPACIDAD (OPACITY)

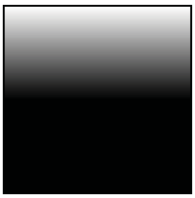


Imagen Original

Transparencia al 50%

Transparencia al 100%

Ejemplos:

```
div { filter: opacity(1);   /* Imagen izquierda */ }
div { filter: opacity(0.5); /* Imagen central */ }
div { filter: opacity(0);   /* Imagen derecha */ }
```

4.2.5.7 SATURACIÓN (SATURATE)

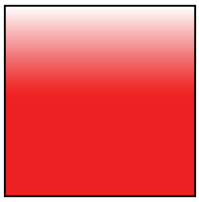
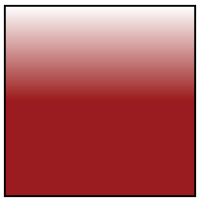


Imagen Original



Saturación al 50%

Sin Saturación de Color

Ejemplos:

```
div { filter: saturate(1);   /* Imagen izquierda */ }
div { filter: saturate(0.5); /* Imagen central */ }
div { filter: saturate(0);   /* Imagen derecha */ }
```

4.2.5.8 SEPIA (SEPIA)

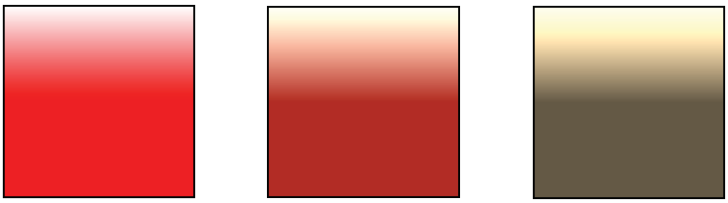


Imagen Original | Sepia al 50% | Sepia al 100%

Ejemplos:

```
div { filter: sepia(0);   /* Imagen izquierda */ }
div { filter: sepia(0.5); /* Imagen central */ }
div { filter: sepia(1);   /* Imagen derecha */ }
```

4.3 TRANSICIONES

4.3.1 Propiedad transition-delay

Ejemplos:

```
div { transition-delay: 1s; }
div { transition-delay: 1.25s; }
div { transition-delay: 25ms; }
```

4.3.2 Propiedad transition-duration

Ejemplos:

```
div { transition-duration: 1s; }
div { transition-duration: 0.75s; }
div { transition-duration: 250ms; }
```

4.3.3 Propiedad transition-property

Ejemplos:

```
div { transition-property: all; }
div { transition-property: transform; }
div { transition-property: width, height, padding; }
```

4.3.4 Propiedad transition-timing-function

Ejemplos:

```
div { transition-timing-function: linear; }
div { transition-timing-function: ease-in-out; }
div { transition-timing-function: cubic-bezier(0.20, 0.1, 0.50, 1); }
```

4.4 ANIMACIONES

4.4.1 Propiedad animation-delay

Especifica el retraso de empiece, es decir, el tiempo que se debe esperar para iniciar la animación. Entre sus posibles valores podemos encontrar:

Ejemplos:

```
div { animation-delay: 1.50s; }
div { animation-delay: 25ms; }
```

4.4.2 Propiedad animation-direction

Ejemplos:

```
div { animation-direction: reverse; }
div { animation-direction: alternate; }
```

4.4.3 Propiedad animation-duration

Ejemplos:

```
div { animation-duration: 1s; }
div { animation-duration: 1.25s; }
div { animation-duration: 25ms; }
```

4.4.4 Propiedad animation-fill-mode

Ejemplos:

```
div { animation-fill-mode: both; }
div { animation-fill-mode: forwards; }
div { animation-fill-mode: backwards; }
```

4.4.5 Propiedad animation-iteration-count

Ejemplos:

```
div   { animation-iteration-count: infinite; }
span  { animation-iteration-count: 3; } aside
{ animation-iteration-count: 3; }
```

4.4.6 Propiedad animation-name

Ejemplos:

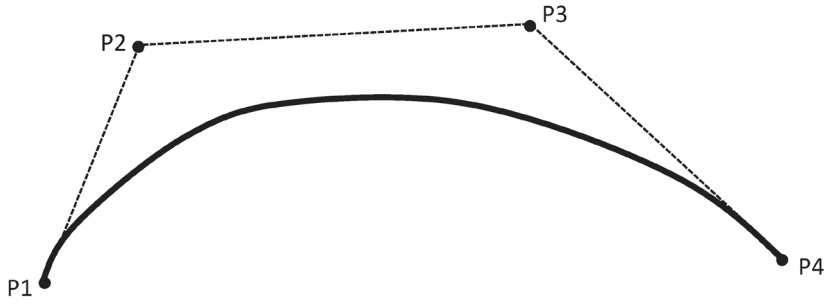
```
div   { animation-name: ejemplo1; }
span  { animation-name: FadeIn; }
aside { animation-name: movetotop; }
```

4.4.7 Propiedad animation-play-state

Ejemplos:

```
div       { animation-play-state: running; }
div:hover { animation-play-state: pause; }
```

4.4.8 Propiedad animation-timing-function



- **EASE:**

Ejemplos:

```
div { animation-timing-function: linear; }
div { animation-timing-function: ease-in-out; }
div { animation-timing-function: cubic-bezier(0.20, 0.1, 0.50, 1); }
```

4.4.9 Regla keyframes

```
@keyframes desplazamiento-lento-rapido {
    from { top: 0; }
    to   { top: 100px; }
}
@keyframes desplazamiento-lento-rapido {
     0% { top: 0; }
    50% { top: 30px; }
   100% { top: 100px; }
}
```

```
@keyframes desplazamiento-lento-rapido {
     0% { top: 0; }
    50% { right: 50px; }
   100% { top: 100px; }
}
```

4.4.10 Ejemplos resueltos

4.4.10.1 EJEMPLO COMPLETO DE TRANSICIÓN

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            .box { width: 100px; height: 100px; background: #000000;
                  transition: width 2s ease-in-out;
            }
            .box:hover { width: 200px; height: 100px; }
        </style>
    </head>
    <body>
        <h1>The transition Property</h1>
        <p>Posiciona el ratón encima del cuadro negro</p>
        <i class=”box”></i>
    </body>
</html>
```

4.4.10.2 EJEMPLO COMPLETO DE ANIMACIÓN

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            .typewriter h1 {
                color: #fff;
                font-family: arial;
                font-size: 1.4rem;
                overflow: hidden;
                border-right: .15em solid #333;
```

```
                white-space: nowrap;
                margin: 0 auto;
                letter-spacing: .2em;
                width: auto;
                display: inline-block;
                animation: typing 3.5s steps(30, end),
                           blink-caret .5s step-end infinite;
            }

            @keyframes typing { from { width: 0 } to { width: 100% } }
            @keyframes blink-caret {
                from, to { border-color: transparent }
                50%      { border-color: #333 }
            }
        </style>
    </head>
    <body>
        <div class=”typewriter”>
            <h1>Esto es un ejemplo de animación typewriter</h1>
        </div>
    </body>
</html>
```

4.4.10.3 EJEMPLO COMPLETO DE EFECTO Y ANIMACIÓN

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            @import url(https://fonts.googleapis.com/css?family=VT323);

            *   { box-sizing: border-box; }
            body{ background: #000; color: #fff;
                     font-family: ‘VT323’; font-size: 24px;
                     line-height: 1.2; -webkit-font-smoothing: none; }
            h1  { font-size: 4rem; margin: 0; text-transform: uppercase; }
            h1 span{ float: right; }
            p      { margin-bottom: 0; color : #ff2022; font-size: 64px;
                     fl     right; text-transform: uppercase; width: auto;
```

```
                position: relative; left: 100%;
                animation: move 20s steps(2000, end) infinite;
                white-space: nowrap; }
        a, a:visited { border-bottom: 0.2rem solid #ffec80;
                       color: #ffec80;
                       text-decoration: none; }
        a:active, a:focus, a:hover{
            background-color: #ffec80;
            color: #333; }

        .piece { display: block; height: 100%; overflow: hidden;
                left: 0; top: 0; width: 100%; }
        .noclick { pointer-events: none; }
        .frame   { background-color: #181818; border-radius: 10px;
                   padding: 15px; height: auto; position: absolute;
                   top: 2%; left: 2%; width: 96%;
                   pointer-events: none; }
        .emblem1 { display: none; position: absolute; left: 50%;
                   bottom: 0; font-size: 2rem; height: 3rem; width:
                   3rem; text-align: center; color: white;
                   background: -webkit-linear-gradient(#fff, #555);
                   -webkit-background-clip: text;
                   -webkit-text-fill-color:  transparent;
                   text-shadow: 0 0.1rem 0rem rgba(0,0,0,0.4),
                                0 0 2rem rgba(0,0,0,0.8);
                   transform: translate(-50%); }
        .output1 { animation: output 10ms infinite;
                   background-color: #333;
                   overflow: scroll; position: absolute;
                   padding: 3rem 2rem; pointer-events: auto;
                   text-shadow: 0rem 0.2rem 1rem
                                lighten(var(custom-bg), 20%);
                   z-index: -1; }

        .scanlines { background: linear-gradient(to bottom,
                     rgba(255,255,255,0), rgba(255,255,255,0) 50%,
                     rgba(0,0,0,0.2) 70%, rgba(0,0,0,0.6));
                     background-size: 100% 0.3rem; border-radius: 2rem;
                     position: absolute; }

        .glow1 { animation: glow 60s infinite;
```

```
                        background: radial-gradient(circle at center,
                                    rgba(27,212,89,1) 0%,
                                    rgba(27,212,89,0.88) 58%,
                                    rgba(21,235,92,0.57) 80%,
                                    rgba(19,94,29,0.27) 93%,
                                    rgba(10,23,12,0) 100%);
                        opacity: 0.15; pointer-events: none;
                        position: fixed; }

            @keyframes move   { from { left: 200vw } to { left: -100vw } }
            @keyframes output {   0% { opacity: 0.9; }
                                 50% { opacity: 1; } }
            @keyframes glow   {   0% { opacity: 0.1; }
                                 50% { opacity: 0.2; } }

        </style>
    </head>
    <body class="noisy">
        <div class="frame">
            <div class="piece output">
                <h1>Madrid N.Min <span>1min</span></h1>
                <p>Próximo tren efectuará parada en Chamartín, Ramón y
Cajal, Pitis y las Rozas</p>
                <div class="piece scanlines noclick"></div>
                <div class="piece glow noclick"></div>
            </div>
        </div>
    </body>
</html>
```

4.5 EFECTOS CON TEXTOS

4.5.1 Efectos estéticos y con movimiento

```
<h2>Cabecera H2</h2>
```

```
h2 {
    display: block;
    text-align: center;
    color: #000 !important;
    text-shadow: 0px 20px #d6d6d6;
}
```

Y el resultado debería ser algo como:

```
h2 {
    display: block;
    text-align: center;
    color: #000 !important;
    position: relative;
}
h2::after {
    content: “Cabecera H2”;
    position: absolute;
    left: 2px;
    top: 20px;
    width: 100%;
    height: 100%;
    transform: skew(15deg);
    color: #dfe1e4;
}
```

Y el resultado debería ser algo como:

```
@keyframes move {
    0%   { transform: skew(-15deg);}
    50%  { transform: skew(15deg);}
    100% { transform: skew(-15deg);
}

/* Para el caso con text-shadow */
h2 {
    display: block;
    text-align: center;
    color: #000 !important;
    position: relative;
    animation-name: move;
    animation-timing-function: cubic-bezier(0.4, 0, 1, 0.8);
    animation-iteration-count:  infinite;
    animation-duration: 3s;}
}

/* Para el caso con el pseudo elemento after */
h2::after {
    content: “Cabecera H2”;
    position: absolute;
    left: 2px;
    top: 20px;
    width: 100%;
    height: 100%;
    transform: skew(15deg);
    color: #dfe1e4;
    animation-name: move;
    animation-timing-function: cubic-bezier(0.4, 0, 1, 0.8);
    animation-iteration-count:  infinite;
    animation-duration: 3s;}
}
```

A continuación mostramos el código para realizar algunos efectos comunes en Internet a través de text-shadow.

4.5.1.1.1 Sombra sólida

```
h2.solida{
    text-shadow: -1px 1px #333, -2px 2px #333,
                 -3px 3px #333, -4px 4px #333, -5px 5px #333;
    color: #060;
}
```

El resultado debería ser algo como:

4.5.1.1.2 Efecto de fuego

```
h2.fuego{
    text-shadow: 0 3px 20px red, 0 0 20px red,
                 0 0 10px orange, 4px -5px 6px yellow,
                -4px -10px 10px yellow, 0 -10px 30px yellow
    color: #666;
}
```

El resultado debería ser algo como:

4.5.1.1.3 Texto grabado

```
h2.grabado {
    background-color: #414141;
    color: #ddd;
    padding: 10px;
    text-shadow: 1px 1px #555, -1px -1px #333;
}
```

El resultado debería ser algo como:

4.5.1.1.4 Texto en relieve

```
h2.relieve {
    background-color: #414141;
    color: #ddd;
    padding: 10px;
    text-shadow: -1px -1px #555, 1px 1px #333;
}
```

El resultado debería ser algo como:

4.5.1.1.5 Efecto rayado

```
h2.multiple{
 text-shadow: 1px 1px #ccc, 2px 2px #aaa, 3px 3px #888, 4px 4px #666
}
```

El resultado debería ser algo como:

4.5.1.1.6 Efecto neón

```
h2.neon{
    background: #000;
    color: #fff;
    font-family: “Open Sans”;
    font-size: 48px;
    height: 180px;
    padding-top: 55px;
    font-weight: 900;
    text-shadow: 0 0 7px #fff,
                 0 0 10px #fff,
                 0 0 20px #fff,
                 0 0 40px #ff2dff,
                 0 0 60px #ff2dff,
                 0 0 80px #ff2dff,
                 0 0 120px #ff2dff,
                 0 0 150px #ff2dff;
}
```

El resultado debería ser algo como:

```
h2.neon{
    animation: parpadeo 5s infinite alternate;
    background: #000;
    color: #fff;
    font-family: “Open Sans”;
    font-size: 48px;
    height: 180px;
    padding-top: 55px;
    font-weight: 900;
}
@keyframes parpadeo {
    0%, 20%, 26%, 28%, 55%, 70%, 100% {
        text-shadow: 0 0 7px #fff, 0 0 10px #fff,
                     0 0 20px #fff, 0 0 40px #ff2dff,
                     0 0 60px #ff2dff, 0 0 80px #ff2dff,
                     0 0 120px #ff2dff, 0 0 150px #ff2dff;
    } 25%, 27%, 54.5% {
        text-shadow: none;
    }
}
```

4.5.3 Adecuación de los efectos a la página web

```
<button class="toggle-collapse">Mostrar / Ocultar Menú</button>
```

```
<nav class="offcanvas">
    <h3>Menú Principal</h3>
    <ul>
        <li><a href="#">Inicio</a></li>
        <li><a href="#">Servicios</a></li>
        <li><a href="#">Contactar</a></li>
    </ul>
</nav>
```

```
<style>
    nav.offcanvas{
        position: fixed;
        top: 0;
        left: -300px;
        width: 300px;
        height: 100%;
        transition: left 0.4s ease;
```

```
        z-index: 99;
    }
</style>
```

```
<style>
    nav.offcanvas.active{
        left: 0;
    }
</style>
```

```
<script type="text/javascript">
    document.querySelector(".toggle-collapse").onclick = function(e){
        e.target.classList.toggle("active");
        document.querySelector("nav.offcanvas").classList.
toggle("active")
    }
</script>
```

```
if(e.target.classList.contains("active")){
    e.target.innerHTML = "Ocultar Menú"
} else {
    e.target.innerHTML = "Mostrar Menú"
}
```

4.6 CREACIÓN DE MAPAS

4.6.1 Incrustación de etiqueta IMG

```
<img src=”ruta/a/imagen.jpg” alt=”Descripción de la imagen”>
```

4.6.2 Agregado del atributo usemap

```
<img src=”ruta/a/imagen.jpg”
     alt=”Descripción de la imagen”
     usemap=”#nombreMapa”>
```

4.6.3 Crear el archivo de mapa de imagen

4.6.4 Definir las áreas interactivas

```
<map name="nombreMapa">
    <area shape="rect"
          coords="x1,y1,x2,y2"
          href="enlace.html"
          alt="Descripción del área">
    <!-- Otras áreas interactivas -->
</map>
```

4.6.5 Guardar y subir el archivo

4.7 COMO ARRASTRAR Y COLOCAR IMÁGENES

4.7.1 Crear el marcado HTML

```
<div id="contenedor" class="contenedor">
   <div id="arrastre" class="arrastre">
      Arrastra y suelta la imagen aquí
   </div>
   <input type="file" id="seleccionador" accept="image/*">
</div>
```

4.7.2 Estilizar con CSS

```
.contenedor {
  width: 300px;
  height: 300px;
  border: 2px dashed #ccc;
  text-align: center;
  padding: 20px;
}

.arrastre {
  font-size: 16px;
  color: #999;
}

.arrastre:hover {
  cursor: pointer;
  color: #666;
}
```

4.7.3 Agregar el script de JavaScript

```
const contenedor = document.getElementById('contenedor');
const arrastre = document.getElementById('arrastre');
const seleccionador = document.getElementById('seleccionador');

arrastre.addEventListener('dragover', (e) => {
   e.preventDefault();
});

arrastre.addEventListener('drop', (e) => {
   e.preventDefault();
   const archivo = e.dataTransfer.files[0];
   mostrarImagen(archivo);
});

seleccionador.addEventListener('change', (e) => {
      const archivo = e.target.files[0];
      mostrarImagen(archivo);
});
```

```
function mostrarImagen(archivo) {
      const lector = new FileReader();
      lector.onload = function(e) {
         const imagen = new Image();
         imagen.src = e.target.result;
         contenedor.innerHTML = ‘’;
         contenedor.appendChild(imagen);
      }
      lector.readAsDataURL(archivo);
}
```

5

OPTIMIZACIÓN Y LIMITACIONES DEL DISEÑO

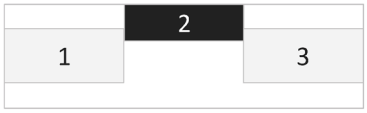
5.1 REPARTICIÓN DE ESPACIOS DENTRO DE UNA PÁGINA

Código CSS

```
html,
body { margin: 0; padding: 0; font-family: ‘Roboto’, sans-serif;
       display: block; font-size: 14px;
}

table         { border: 1px solid rgba(0,0,0,0.2); border-spacing: 2px;
                margin: 10px 0; }

table caption { background: #000000; color: #fff; font-size: 1.0rem;
                font-weight: bold; line-height: 1.5; padding: 0;
                text-transform: uppercase; }

table td,
table th      { border: 1px solid rgba(0,0,0,0.2); border-spacing: 0;
                font-size: 1rem; padding: 5px; text-align: left;
                margin: 2px 0; white-space: nowrap; }

table thead th:nth-child(6),
table tbody td:nth-child(6){ text-align: right; }

@media screen and (max-width: 620px) {
    table { width: 100%; }

    thead { display: none; }

    tr td:first-child { background: #f0f0f0; font-weight: bold; }

    tbody td { display: block; text-align: right; }

    tbody td:before { content: attr(data-field); display: block;
                      fl     left; font-weight: bold; padding: 0 10px 0 0;
                      text-align: left; width: auto; }
}
```

5.1.2.1.1 Propiedad allign-self

Especifica la alineación determinada para un elemento que está dentro de un contenedor flexible. Entre sus posibles valores podemos encontrar:

AUTO: indica que la alineación es inherente y que debe heredarse de la propiedad ALIGN-ITEMS definida en su contenedor. Es el valor por defecto.	
BASELINE: indica que el elemento debe estar posicionado en la línea base del contenedor flexible.	
CENTER: indica que el elemento debe estar posicionado en la parte central del contenedor flexible.	
FLEX-END: indica que el elemento debe estar posicionado al final del contenedor flexible.	
FLEX-START: indica que el elemento debe estar posicionado al principio del contenedor flexible.	
STRECTCH: indica que el elemento se debe ajustar al alto del contenedor para rellenarlo.	

NOTA

La propiedad ALIGN- ITEMS sólo tendrá algún efecto cuando el modo de visualización (DISPLAY) sea FLEX.

5.1.2.1.2 Propiedad flex

```
li  { flex: 1 1 auto; } /* FLEX-GROW FLEX-SHRINK FLEX BASIS */
```

```
p   { flex: 1 100%; }  /* FLEX-GROW FLEX-BASIS */
p   { flex: 1 1; }     /* FLEX-GROW FLEX-SHRINK */
p   { flex: 1; }      /* FLEX-GROW */
p   { flex: 100%; }   /* FLEX-BASIS */
```

5.1.2.1.3 Propiedad flex-basis

5.1.2.1.4 Propiedad order

5.1.2.2 CREACIÓN DE FLEXBOX RESPONSIVE

```
<div class="flexbox">
    <div class="caption">EJEMEPLO DE TABLA CON FLEXBOX CSS</div>

    <div class="row header">
        <div class="col">ID</div>
        <div class="col">Empresa</div>
        <div class="col">F. Movimiento</div>
        <div class="col">Tipo</div>
        <div class="col">Concepto</div>
        <div class="col">Importe</div>
        <div class="col">Estado</div>
```

```
    </div>

    <div class="row">
        <div class="col">1</div>
        <div class="col">Consultores SA</div>
        <div class="col">30-12-2019</div>
        <div class="col">Ingreso</div>
        <div class="col">Nómina</div>
        <div class="col">+1268.00 €</div>
        <div class="col">Efectuado</div>
    </div>

    <div class="row">...</div>
    ...
</div>
```

Código CSS

```
.flexbox .caption { display: block; text-align: center; font-weight:
600;
                    background: #000; color: #fff; }
.flexbox          { border: 1px solid #ccc; }
.flexbox .row     { display: flex; width: 100%; max-width: 100%;margin:
0;}
.flexbox .col     { border: 1px solid #ccc; display: flex;
                    flex-flow: column nowrap; justify-content: flex-start;
                    align-items: flex-start; flex: 1 1 100%; margin: 1px;
                    padding: 0 5px; max-width: calc(100% / 7); }
.flexbox .header .col { font-weight: 600; }
```

5.1.3 Diseño basado en cuadrículas (Grid Layout)

5.1.3.1.1 Propiedades grid-template-rows y grid-template-columns

```
.grid {
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr 1fr 1fr 1fr;
    grid-template-rows: 1fr 1fr 1fr 1fr;
    gap: 5px;
}
```

5.1.3.1.2 Propiedades grid-row-start y grid-row-end , grid-column-start, grid-column-end

```
.large-item {
  grid-column-start: 2;
  grid-column-end: five;
  grid-row-start: row1-start;
  grid-row-end: 3;
}
```

5.1.3.1.3 Propiedad align-items y justify-items

Especifican cómo se deben distribuir los elementos horizontal y/o

```
.grid {
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr 1fr 1fr 1fr;
    grid-template-rows: 1fr 1fr 1fr 1fr;
    gap: 5px;
    align-items: center;
    justify-items: center;
}
```

5.1.3.1.4 Función repeat y las palabras clave

```
.grid {
    display: grid;
    grid-template-columns: repeat(7, 1fr);
    grid-template-rows: repeat(4, 1fr);
    gap: 5px;
}
```

```
.grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
    grid-template-rows: minmax(max-content, 1fr);
    gap: 5px;
}
```

```
<div class="grid">
    <div class="caption">EJEMEPLO DE TABLA CON FLEXBOX CSS</div>

    <div class="row">
        <div class="col">ID</div>
        <div class="col">Empresa</div>
        <div class="col">F. Movimiento</div>
        <div class="col">Tipo</div>
        <div class="col">Concepto</div>
        <div class="col">Importe</div>
        <div class="col">Estado</div>

        <div class="col">1</div>
        <div class="col">Consultores SA</div>
        <div class="col">30-12-2019</div>
        <div class="col">Ingreso</div>
        <div class="col">Nómina</div>
        <div class="col">+1268.00 €</div>
        <div class="col">Efectuado</div>

        <div class="col">2</div>
        ...
    </div>
</div>
```

Después, sólo necesitaremos definir el sistema de grid a la clase **.ROW** y unos cuantos estilos adicionales:

Código CSS

```
.grid .row     { display: grid; grid-template-columns: repeat(7, 1fr);
                 grid-template-rows: minmax(max-content, 1fr);
                 padding: 1px 1px; border: 1px solid #ccc; }
.grid .caption { display: block; text-align: center; font-weight: 600;
                 background: #000; color: #fff; }
.grid .col     { border: 1px solid #ccc; margin: 1px; padding: 0 5px; }
.grid .row .col:nth-child(-n+7) { font-weight: 600; }
```

5.2 INSERCIÓN DE UN BACKGROUND

5.2.2 Adaptación receptiva y adaptativa

```
<style>
    .banner {
        background-image: url(imagen-640x360.jpg);
        border-bottom: 1px solid rgba(0,0,0,0.1);
```

```
        color: #fff;
        display: block;
        height: 100vh;
        position: relative;
        text-align: center;
        width: 100%;
    }

    @media (min-width: 800px) {
        .banner { background-image: url(imagen-1280x720.jpg); }
    }

    @media (min-width: 1400px) {
        .banner { background-image: url(imagen-1920x1080.jpg); }
    }

    @media (min-width: 2000px) {
        .banner { background-image: url(imagen-2560x1440.jpg); }
    }
</style>

<div class=”banner”><div>
```

5.3 CREACIÓN DE HIPERVÍNCULOS EN REGIONES DE LA PANTALLA

5.3.1 Texto con hipervínculos

:

```
<a href=”pagina.html”>Enlace a otra página</a>
```

5.3.2 Imágenes con hipervínculos

:

```
<a href=”pagina.html”>
      <img src=”imagen.jpg” alt=”Descripción de la imagen”>
</a>
```

5.3.3 Mapas de imagen

```
<img src=”plano.png” alt=”Plano de la casa” usemap=”#planomap”>
<map name=”planomap”>
      <area shape=”rect”
             coords=”0,0,100,100”
             href=”pagina1.html”
             alt=”Área 1”>
      <area shape=”rect”
             coords=”100,0,200,100”
             href=”pagina2.html”
             alt=”Área 2”>
</map>
```

5.3.4 CSS para estilizar hipervínculos

```
a {
        color: blue;
        text-decoration: underline;
}

a:hover {
        color: red;
}
```