

Tema 1: Planificación de interfaces gráficas

¿Qué aprenderás?

- Reconocer los principios básicos de la comunicación visual.
- Analizar y seleccionar colores y tipografías adecuadas.
- Crear una guía de estilos.
- Desarrollar prototipos para el diseño de interfaces estructurando los distintos componentes del mapa web.

¿Sabías que...?

- En 1891 William Morris creó Kelmscott, la primera empresa dedicada al diseño gráfico.
- El color amarillo de *Los Simpson* fue escogido para llamar la atención de los espectadores mientras hacían *zapping*.
- Woody Allen utiliza la tipografía Windso en todas sus películas.



1. Planificación de interfaces gráficas

1.1. La comunicación visual

La vista es uno de los principales sentidos que utilizamos consciente e inconscientemente para comunicarnos con nuestro alrededor. Cuando observamos una nueva interface gráfica de una aplicación, de forma inconsciente valoramos muy acertadamente si está bien hecho. Por desgracia el diseño consciente de una nueva interfaz o estructura visual va viciando nuestra percepción inconsciente.

Para diseñar conscientemente una nueva estructura visual que transmita la información que queramos tenemos que tener presentes cuales son los elementos básicos que conforman cualquier diseño.

1.1.1. Elementos básicos del diseño

Si analizamos una obra visual descomponiéndola en los elementos que la constituyen podemos observar cuales esos elementos básicos y como se han utilizado para lograr un determinado efecto. Los elementos básicos que podemos utilizar para los diseños son:

- a) **El Punto:** es la unidad más simple de comunicación visual. Una gran agrupación de puntos de distintos colores o de un mismo color pero con distintas separaciones puede crear la ilusión de un color o un tono. El punto puede ser utilizado como referencia, para marcar un espacio y atraer la atención del ojo o guiar el ojo.



- b) **La Línea:** cuando el espacio entre una serie de puntos es imperceptible se forma una línea. La línea es el elemento esencial del dibujo, nos permite representar aquello que solo existe en nuestra imaginación. Dependiendo de lo que queramos expresar, las líneas pueden ser rectas y marciales o desenfrenadas y expresivas.

- c) **Dirección:** cada línea tiene una dirección, asimismo todos los elementos que conforman un diseño provocan una dirección global en el diseño. Cada dirección puede provocar distintas percepciones.

- a. Vertical: sensación de altura y cantidad.
- b. Horizontal: sensación de estabilidad, bienestar y maniobrabilidad
- c. Diagonal: sensación de crecimiento (si la línea sube), descenso (si la línea baja), inestabilidad y energía.



d. Curva: sensación de repetición, movimiento y calor.

d) El **contorno**: distintas líneas combinadas en distintas direcciones pueden formar un contorno. Hay tres contornos básicos que podemos ir combinando para obtener todas las formas que podamos imaginar. A cada uno de los contornos se le atribuyen distintos significados a partir de las propias percepciones subjetivas y las direcciones de las líneas que los forman:

a. Cuadrado: recto y serio se le asocian significados de seriedad, rectitud, estabilidad, cantidad.



b. Triángulo: enérgico y activo se le asocian significados de acción, conflicto y tensión.

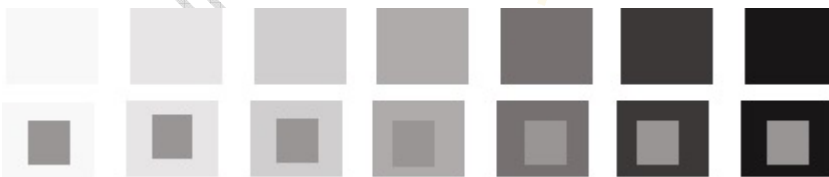


c. Círculo: protector, se le asocian significados de infinitud, calidez y protección.



e) **Tono**: es la cantidad de luz con la que vemos un color. Entre la luz y la oscuridad real hay cientos de grados tonales distintos que como seres humanos no somos capaces de apreciar, por eso se suelen usar más de trece grados tonales distintos para derivar del blanco hacia el negro.

Superponer dos tonos distintos ayuda a percibir la diferencia de un tono y el resto.



El tono es clave para la sensación de profundidad, distancia y luz.



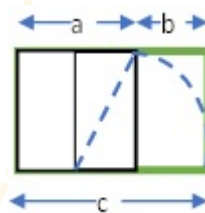


- f) **Color:** el color tiene una gran afinidad con las emociones fruto de la asociación que hacemos con los colores de nuestro entorno (y nuestra sociedad). Por ello aunque algunas emociones son comunes entre distintas personas, otras pueden variar según la cultura y el entorno en el que haya vivido la persona. Como el color es uno de los elementos primordiales en el desarrollo de interfaces web, le dedicaremos más a delante un apartado propio.
- g) **Textura:** la textura es una repetición de un mismo patrón sobre una superficie que en muchas es asociada a una percepción táctil. Las texturas deben utilizarse con sumo cuidado y muy escasamente porque dificultan la comprensión de las formas. Por ejemplo, utilizar texturas como fondo de un texto dificultará la legibilidad del texto.
- h) **Escala:** La escala es el proceso en el que un elemento visual define el tamaño de otro elemento visual comparándose entre ellos. Por ello refiriéndonos a la escala un elemento será grande o pequeño según la composición con otros elementos, no según sus medidas. En el siguiente ejemplo el elemento A es grande respecto al B, pero es pequeño respecto al C.



Un mismo elemento representado en distintas escalas obligará al cerebro a asignar distintos valores de distancia, jerarquía o prioridad. Una sobreutilización de distintas escalas en una interface puede resultar agotador para el cerebro.

Utilizar fórmulas proporcionales sobre las que basar una escala puede ayudar a mostrar una interface más unificada y apetecible. Una de las más famosas es “la proporción áurea” que escala todos los elementos a partir de un “rectángulo áureo” obtenido de utilizar la diagonal de la mitad de cuadrado como radio para ampliar las dimensiones del cuadrado transformándolo en un rectángulo. Consiguiendo la proporción $a:b = c:a$



- i) **Dimensión:** en el mundo real físico podemos sentir y ver la dimensión como el volumen de las cosas. Pero en las representaciones bidimensionales el volumen será una ilusión por medio de la perspectiva, el tono y la escala. En el diseño de interfaces utilizaremos la dimensión sobre todo para mostrar información crítica al usuario necesaria para que siga con la aplicación. Como por ejemplo los avisos de utilización de cookies.



- j) **Movimiento:** el movimiento es una de las fuerzas visuales más enérgicas a la que se expone el ser humano. Incluso en imágenes estáticas podemos utilizar distintas técnicas para dotar de la energía del movimiento.

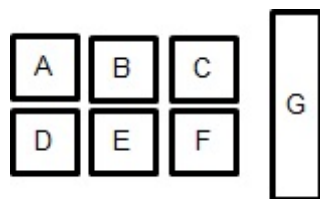
1.2. Principios básicos del diseño

Los principios básicos de diseño intentan marcar unas reglas para diseñar de forma consciente proyectos gráficos con una apariencia atractiva y equilibrada utilizando los elementos básicos del diseño.

Los elementos básicos que podemos utilizar para los diseños son:

- a) **Balance:** indica la compensación o equilibrio de los elementos en una estructura gráfica respecto a un eje de composición. Un balance simétrico transmite fortaleza, estabilidad y formalidad. Un balance asimétrico transmite por el contrario informalidad, vanguardia y energía.
- b) **Contraste:** nos ayudará a diferenciar los distintos contornos fácilmente. Se puede producir por color, forma, escala o dirección. La falta de contraste puede ser un grave problema porque va a provocar que no se diferencien los elementos que componen el diseño. Podemos utilizar el contraste para indicar la jerarquía en la información, por ejemplo dotando de mayor tamaño a los titulares o marcando en negrita las palabras clave.
- c) **Alineación:** tendemos a crear conexiones entre los distintitos contornos que comparten alineación. De ésta forma podremos agrupar la información de los distintos elementos visuales si éstos están alineados o desordenarlos si los desalineamos. En general vamos a preferir alinear unos elementos con los otros para que estén ligados entre ellos y sea más fácil comprender y leer la interfaz.
- d) **Proximidad:** como la alineación, la proximidad establece relaciones entre los distintos contornos. Pero además la escala en la proximidad va a indicar el grado de relación entre dos elementos. Por ello deberíamos poner más juntos aquellos elementos que queramos relacionar más y dejar lejos aquellos elementos que no tengan apenas alguna relación.
- e) **Repetición:** la repetición de contornos básicos reforzará la relación entre ellos. Si además se combina con una misma alineación y alta proximidad su relación aumentará generando un nuevo elemento visual.

Como ejemplo la siguiente repetición, alineación y proximidad agrupa claramente los elementos de la A a la F.



Ejemplo de repetición, alineación y proximidad



1.3. Colores

Una de las decisiones más importantes al diseñar una interfaz va a ser la elección de los colores y sus combinaciones. Es muy importante que elijamos los colores que van a formar una interfaz lo hagamos de forma consciente teniendo presente las emociones que suelen transmitir y las posibles combinaciones que podemos tener.

1.3.1. Colores y emociones

Las emociones que nos pueden provocar los distintos colores van a depender en gran medida de nuestra cultura y experiencias vividas. Pero se han detectado diversos patrones comunes que podemos aprovechar para influir en las emociones del usuario. A continuación analizaremos los principales colores y las emociones con las que se asocian:

- **Blanco:** en general representa la pureza, la inocencia y la limpieza. No es de extrañar que en culturas Orientales y Africanas se asocie con la muerte en representación de la pureza de las almas que se van.
- **Amarillo:** el color del sol y el oro se asocia con la alegría, calidez y la infancia así como la prosperidad, la felicidad y el poder. En USA se asocia con la cobardía y la temporalidad.
- **Verde:** el color de las plantas se asocia con la vida, la naturaleza, la relajación, la juventud y la esperanza. También es el color de los celos y envidia.
- **Azul:** el color del agua y el cielo despejado se asocia con la calma, tranquilidad, estabilidad, seriedad, confianza e inteligencia. Es uno de los colores corporativos más utilizados.
- **Rojo:** el color del fuego y la sangre se asocia con la energía, el peligro, la inmediatez, la pasión y hasta cierta agresividad. Es uno de los colores que afecta con mayor intensidad en nuestro comportamiento. En China se asocia con la felicidad y la suerte, mientras que en Egipto se asocia con la muerte.
- **Marrón:** el color de la tierra y los troncos se asocia con la solidez, la seguridad, la protección, lo rústico y el confort.
- **Negro:** el color de la oscuridad se asocia con el misterio y la muerte pero también con la elegancia, la sobriedad, el poder y el clasicismo.
- **Naranja:** formado por la felicidad del amarillo y la energía del rojo, el naranja se asocia con la salud, la juventud, la felicidad, la exaltación y sensualidad.
- **Rosa:** se asocia ampliamente con la dulzura, la delicadeza, la sensibilidad, el cariño, la amistad, y la sexualidad del amor puro.
- **Morado:** se asocia ampliamente con el glamour, la sofisticación y la realeza pero también con el misterio, la nostalgia y espiritualidad.
- **Gris:** es color frío que se asocia a la sobriedad, el luto, la discreción y la profesionalidad.



1.3.2. Combinaciones de colores

Para el diseño de una página crearemos una composición de distintos contornos junto con una combinación de distintos colores. Como norma general intentaremos usar entre 3 y 5 colores a parte del color de fondo (como color de fondo se suele recomendar el blanco). El color principal será el que guíe la web y transmita la mayor sensación. El secundario sirve para matizar el principal y el tercero lo utilizaremos como destacado para resaltar por encima el principal.

La elección de los colores que formen una combinación ha de ir en consenso con el significado que queramos transmitir al público objetivo, pero la combinación de colores también ha de resultar agradable.

En base a la rueda de colores hay distintas armonías que marcan unas directrices para escoger una paleta de colores armónica y que ayude a transmitir distintas emociones:

- **Combinación complementaria:** se forma en utilizar colores situados en lados opuestos en la rueda de colores. La combinación resultante será muy animada y llena de energía:





- **Combinación de complementaria por separado:** parecida a la complementaria ésta combinación se compone de los dos colores adyacentes al complementario:



- **Combinación tríadica:** se forma en seleccionar tres colores equidistantes en la rueda de colores. Ésta combinación proporciona un gran contraste y sensación vivaz. Es importante seleccionar uno de los tres colores como color predominante.

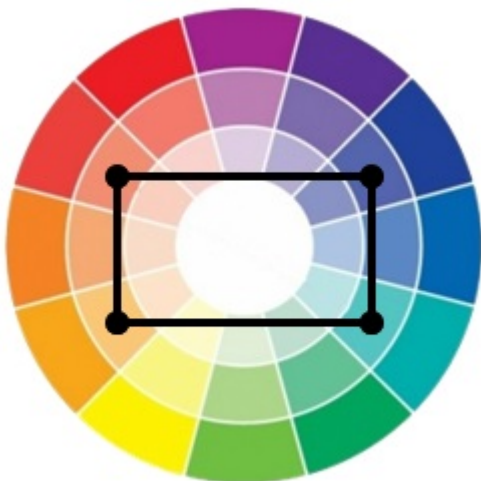




- **Combinación de análogos:** se forma en combinar de 2 a 5 colores ubicados uno al lado del otro en la rueda de colores. Es una combinación que transmite tranquilidad y es muy frecuente encontrarla en la web junto con un color complementario.



- **Combinación tétrada:** se forma al combinar dos conjuntos de colores complementarios. Generan un estilo muy llamativo.



1.4. Tipografía

Al diseñar una interfaz web no tenemos que olvidar que los textos van a ser esenciales para transmitir la información. El estilo que tengan las letras que formen la web puede incidir en la facilidad de lectura y generar distintas emociones. Nos referiremos a estos estilos distintos de letras como “fuentes”.

Cuando nos hablemos en el contexto de tipografía, una fuente es un conjunto de letras con un estilo común. Como desarrolladores nos podemos descargar y utilizar en nuestra página web cualquier fuente, pero si no es una fuente estándar tendremos que configurar los



mecanismos necesarios para que el navegador la descargue de forma automática cuando quiera visualizar la web (ya lo veremos al trabajar con los CSS).

Existen dos grandes tipos de fuentes tradicionalmente usadas para la transmisión de información: Serif y Sans Serif.



- **Serif:** cada letra tiene un remate como adorno. Utilizadas sobre un soporte impreso el remate ayuda a ligar las letras haciendo que el mensaje sea más fácilmente legible. Por ello también son aconsejadas para la escritura de párrafos. Su uso en pantallas dependerá de la calidad de la pantalla. Si la pantalla tiene una baja densidad en píxeles los remates no se van a ver correctamente y entorpecerán la lectura. Por suerte en la mayoría de pantallas modernas se superan los 220PPI (píxeles por pulgada) y ya no se produce ésta situación.
- **Sans Serif:** las letras no tienen ningún remate como adorno. En soporte impreso son aconsejadas para los títulos. En pantallas se deben utilizar cuando la calidad de las mismas sea muy baja con resoluciones inferiores a 220PPI. En particular destacan por su fácil lectura las fuentes Verdana y Calibri.

1.5. Mapa de navegación

Cuando empecemos a diseñar un sitio web vamos a tener que pensar en el contenido que va mostrar, como lo vamos a organizar y cómo vamos a acceder a él. El mapa de navegación de un sitio web presenta un esquema de las distintas páginas del sitio y sus relaciones.

Tenemos que diferenciar entre:

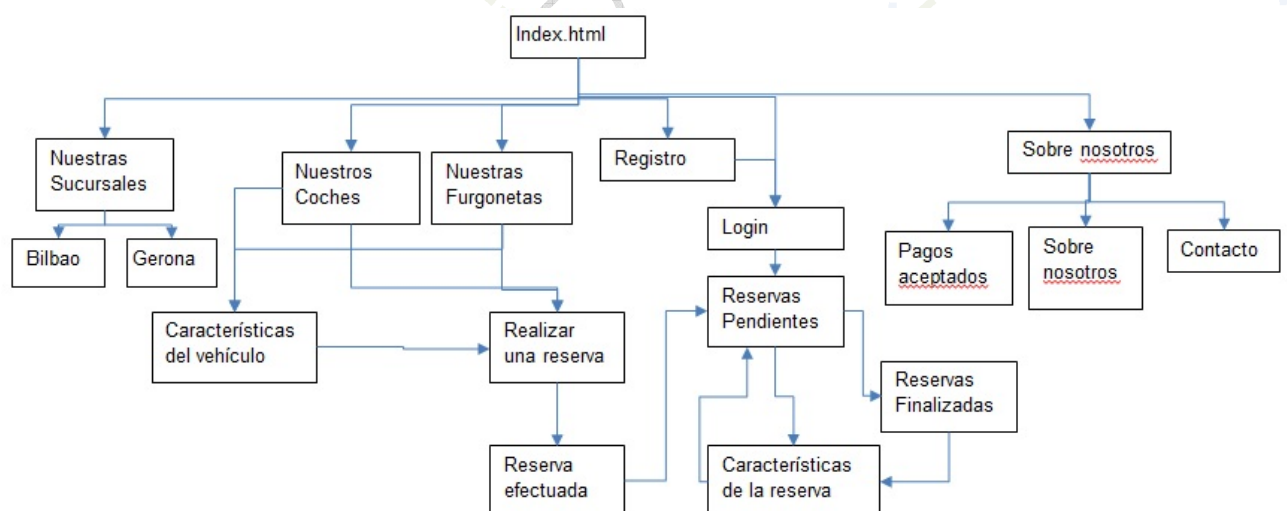
- Mapas visuales de navegación del sitio web: son representaciones gráficas de los contenidos de un sitio web agrupados por páginas y relacionados entre ellos. Su objetivo es mostrar gráficamente de qué contenidos web se compone un sitio y como se pueden acceder a ellos.
- Mapas de navegación HTML: se recomienda añadir en la parte inferior de las páginas de nuestro sitio web un índice de los contenidos generales que ofrece nuestro sitio web. Generalmente agrupado por temáticas y con links directos a las distintas secciones.



- Sitemaps XML: es una estructura XML, desarrollada por Google, en el que estructuraremos las páginas web de un sitio web y sus interrelaciones. Su principal objetivo es facilitar la indexación de los recursos que ofrece un sitio web por parte de los buscadores pudiendo indicar también cuales son los últimos cambios que se han ido produciendo (para que el buscador se centre solo en indexar esos contenidos).

El Mapa visual de navegación ha de indicar como se puede acceder a cada una de las secciones (páginas) del sitio web. Un mapa del sitio web puede ser en alta o baja fidelidad pudiendo incluir absolutamente todos los caminos o señalando solo los caminos principales. Con sitios web complejos suele ser más práctico un mapa en baja fidelidad en el que no incluiremos todos los posibles caminos para acceder a aquellos contenidos recurrentes (links contacto, menú principal, términos de uso), sino que partiremos de la página inicial de nuestro sitio y a partir de ése punto intentaremos jerarquizar la información e indiciar los caminos principales para acceder a los distintos contenidos de la web.

A continuación podemos ver un posible mapa de navegación de baja fidelidad de una web de reservas de coches y furgonetas:



Ejemplo mapa visual de navegación de un sitio web de reservas de coches



El correspondiente mapa de navegación HTML podría ser el siguiente:

SOBRE NOSOTROS	NUESTRAS SUCURSALES	NUESTRA FLOTA	ACCESO
<ul style="list-style-type: none">• Pagos <u>aceptados</u>• Sobre <u>nosotros</u>• Contacto	<ul style="list-style-type: none">• Bilbao• Gerona	<ul style="list-style-type: none">• Nuestros <u>Coches</u>• Nuestras <u>Furgonetas</u>	<ul style="list-style-type: none">• Login• Registro

Ejemplo mapa html de navegación de un sitio web de reserva de coches

Una vez ya sepamos de qué secciones se va a componer la página web y hayamos decidido los colores y tipografía, será el momento de empezar a estructurar cada una de las interfaces y sus componentes.

1.6. Composición de una interfaz web

Deberemos evitar componer directamente en Html y Css la estructura de una página web, hacerlo nos va a gastar mucho tiempo y esfuerzo. Antes de empezar a maquetar la web en HTML y CSS empezaremos siempre con unos esbozos de cada una de las diferentes interfaces que van a formar las páginas del sitio web indicando que componentes la van a formar y como se van a situar. A éstos esbozos los llamaremos Prototipos, Wireframes o Mockups.

1.6.1. Prototipos

Los Prototipos, Wireframes o Mockups son esbozos sin estilos, fuentes ni colores que deben indicar de forma simple los distintos componentes y su función en cada una de las distintas interfaces que formará nuestra web o aplicación.

Su objetivo es múltiple, por una parte nos da una idea global de la web ayudándonos a su desarrollo y a detectar posibles carencias y por otra parte nos permite exponer al cliente rápidamente distintos wireframes que van a ser fácilmente modificables hasta consensuar con el cliente los elementos que van a formar la web y blindar ésa idea frente a futuros cambios que al cliente se le puedan ocurrir.

Los wireframes se pueden desarrollar con cualquier herramienta que se nos ocurra, desde un trozo de papel y un lápiz, pasando por un editor de texto o con alguna herramienta especializada. Lo importante es que os resulte sencillo, rápido y práctico.



El siguiente ejemplo muestra un posible wireframe para una interfaz de una página web.

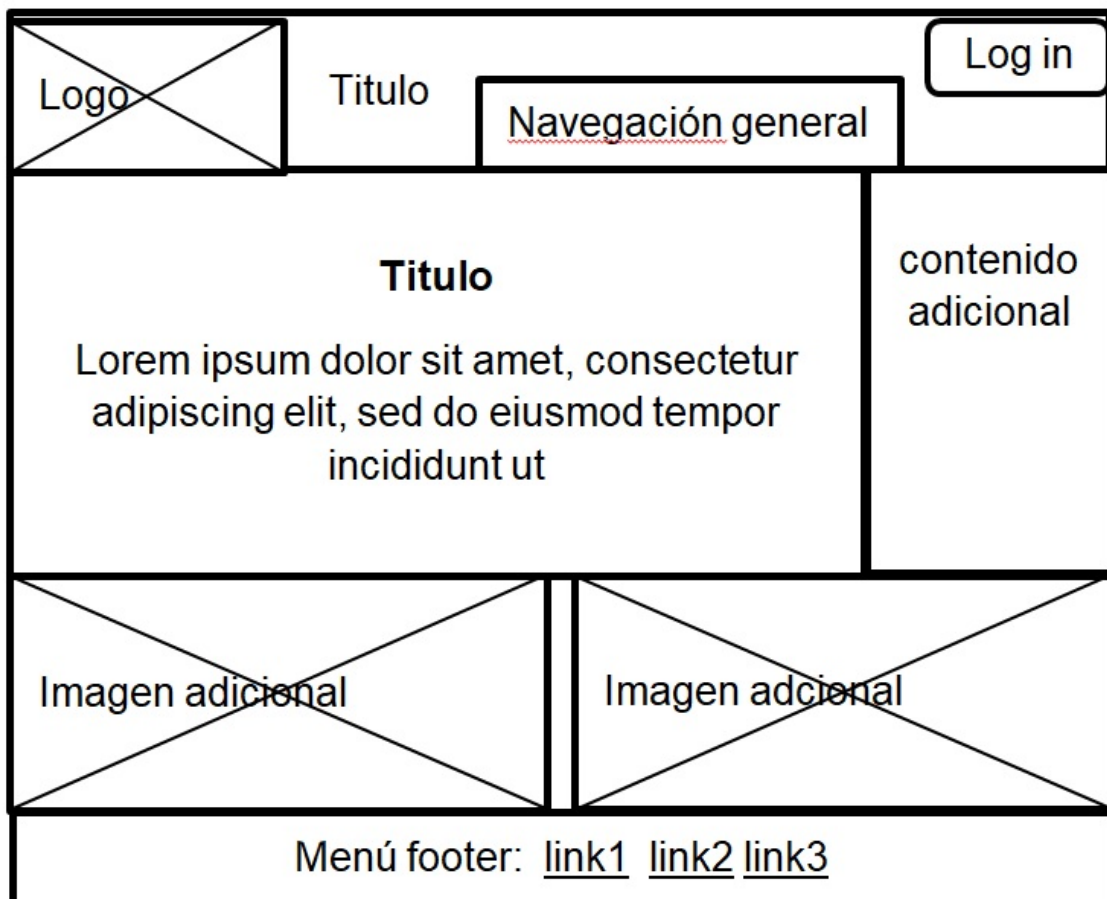
Titulo		Log in
Sing up <input type="text" value="email"/> <input type="button" value="submi"/>	Imagen	
Titulo		
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut		
Menú footer: link1 link2 link3		

Ejemplo de prototipo wireframe

Uno de los aspectos que tendremos que pensar al estructurar la web es qué componentes van a formar parte de ella. Aunque el diseño web está continuamente en evolución algunos componentes son comunes para la mayoría de sitios web. Aunque no son una obligación ni garantía de éxito, sí que nos aseguran que la mayoría de usuarios van a estar familiarizados con y sus funcionalidades.

Los principales componentes de una interfaz web son: La cabecera, el cuerpo, el pie de página y sistema de navegación y contenido adicional.

En el siguiente ejemplo podemos ver un prototipo con una composición web típica:



Prototipo genérico con los principales componentes web

1.6.2. Cabecera

La cabecera identifica el sitio web y contiene las interacciones más genéricas con las que va a poder interactuar el usuario. Suele estar situada en la parte superior de la página web, a mostrar el logo, un título y los botones de login, registro, cambio de idioma así como links a las secciones principales de la web.

Para ayudar a cohesionar el sitio web, se recomienda que la cabecera sea muy parecida entre las distintas páginas que forman el sitio web.

Lo más común es que la cabecera ocupe todo el ancho de la web y el logo se sitúe a la parte izquierda.

1.6.3. Cuerpo

El cuerpo es tradicionalmente lo encontramos debajo de la cabecera y es en donde se muestra al usuario la información principal referente a los contenidos de la página. Suele ser el componente que ocupa un mayor espacio. Puede contener su propio título para identificar el contenido que va a mostrar y también subtítulos si se muestran subcontenidos.



1.6.4. Pie de página

El pie de página se suele situar en la parte inferior de la misma. Suele contener información global de la página como su autoría, información de contacto o información legal como el copyright, sellos de calidad, condiciones de uso o políticas de seguridad.

Cabe destacar que en el cuerpo de la página es posible encontrar pies de página para mostrar información adicional sobre una sección.

1.6.5. Sistemas de navegación

Los sistemas de navegación son los links o menús que permiten a un usuario visitar las distintas secciones de la web o escoger distintas acciones para interactuar con ella.

También podemos encontrar sistemas de navegación formados por una serie de enlaces que describen la ruta de las subsecciones en la que se sitúa la página actual. Éstos menús facilitan que el usuario se ubique dentro de la página web y permite navegar rápidamente por las subsecciones de las que forma parte la página actual.

Los menús pueden ser estáticos o desplegables. Se recomienda utilizar los menús estáticos para indicar los accesos a las acciones o secciones principales y los desplegables para acceder a secciones secundarias.

Con un diseño web adaptable a los distintos tamaños de la pantalla es muy común transformar menús estáticos en desplegables cuando la pantalla sea muy estrecha.

1.6.6. Contenido adicional

La mayoría de páginas web disponen de una a o varias secciones dedicadas a mostrar contenido adicional relacionado con el contenido principal mostrado en el cuerpo de la página. Se suele ubicar en la parte inferior entre el footer y el cuerpo o a la derecha del contenido.

1.7. Guía de estilos

Hasta ahora hemos visto los elementos básicos de los que se va a componer el diseño de la web. Cada vez que creamos una nueva interfaz en un sitio web deberemos asegurarnos que es coherente con el resto de la página. ¿Por qué? Bien, imaginemos que un usuario entra en nuestra página web por primera vez, va a tener que hacer un esfuerzo para aprender la lógica de nuestra página. Si en cada página cambiamos la ubicación de los distintos componentes, el usuario se va a poder sentir abrumado por tener que volver a aprender la interfaz e incluso desorientado al no percibir que sigue en el mismo sitio web.



Para conseguir que toda la web siga un mismo patrón de diseño, deberemos crear una guía de estilos.

Una guía de estilos es conjunto de documentos que define las reglas, posibles interacciones y conjunto de elementos gráficos que deberemos utilizar en cada una de las nuevas interfaces de nuestra web. Su objetivo final es conseguir que todo el sitio web mantenga una misma cohesión visual y funcional incluso en sus futuras ampliaciones o modificaciones.

La guía de estilo la podemos crear en un documento Html definiendo como mínimo los siguientes parámetros:

- **Valores de la marca:** podemos empezar el documento Html con una definición de la misión de la marca así como sus aspectos emocionales, sociales, culturales y valores que quiere transmitir. Indica también a quién va dirigida la marca en mayor y menor medida.
- **Paleta de colores:** Se recomienda que esté compuesta por un mínimo de tres colores y un máximo de cinco. Cada color se debe corresponder con una de los siguientes tres niveles:
 - Color primario: el color predominante en la interfaz.
 - Color secundario: para distinguir distintas áreas de la interfaz.
 - Color destacado: utilizado en momentos puntuales para resaltar mensajes o alertas. Debe contrastar con el primario y el secundario.
- **La tipografía:** En el mismo documento Html añade ejemplos de títulos, subtítulos, links, pies de página, palabras resaltadas, frases citadas y textos largos. Trabaja con distintos colores, subrayados, cursivas, negritas, fuentes y tamaños. Como reglas generales para conseguir una correcta tipografía:
 - Evita usar más de dos tipos distintos de fuentes. Al usuario le va a costar menos percibir la cohesión de tu página web.
 - Evita abusar de las negritas. Las negritas deben indicar algo importante, si todo es importante entonces nada lo es.
 - Evita las fuentes recargadas. Pueden hacer muy difícil la lectura y añaden mucho peso en el texto.
 - Adecúa el tamaño de la fuente a la edad del usuario objetivo.
 - Intenta utilizar el subrayado solamente para marcar los enlaces. Los usuarios están acostumbrados a que los links estén subrayados, por lo que podemos confundir al usuario si los utilizamos de forma poco clara.
 - Evitar utilizar la cursiva. Un exceso provoca mareo.
 - Las mayúsculas son difíciles de leer, por lo que se recomienda usarlas puntualmente.



- **La iconografía:** los iconos son las imágenes gráficas generalmente pequeñas que se suelen utilizar como metáforas de acciones o secciones de la web. Los iconos deben descargar de texto la página web haciéndola más fácilmente legible y comprensible, pero tenemos que escogerlos cuidadosamente teniendo presente que:
 - Los iconos han de ser coherentes también con los valores de la marca, el público objetivo y la paleta de colores.
 - Como van íntimamente relacionados con la cultura, nos debemos asegurar que el público objetivo los va a entender correctamente testeándolos con usuarios reales.
 - Como son completamente subjetivos siempre existe el riesgo que compliquen la interpretación.
 - Como siempre existe el riesgo de la mala interpretación, si los queremos utilizar para operaciones críticas siempre los acompañaremos de un texto.
- **Los formularios:** debemos establecer las estrategias de feed-back con los formularios y en general las partes interactivas de la web. En algunas ocasiones querremos mostrar información cuando el usuario sitúe su ratón encima un elemento, mientras esté escribiendo, cuando haya terminado de escribir o cuando clique en un botón. Debemos definir entonces si el feed-back será un mensaje, un icono o ambos y donde se va a situar.
- **Las imágenes:** debemos incluir imágenes que definan el estilo y los valores de la marca
- **Los botones:** como elementos esenciales de nuestra web deberemos indicar los distintos tipos de botones y sus posibles estados. En general los distintos estados serán: botón activo, botón normal, con el ratón encima o deshabilitado.
- **Espaciado:** debemos indicar también el espaciado en el texto, cabeceras, botones, imágenes, formularios y otros elementos. Para una buena lectura el espaciado entre líneas debería ser de 1.5 puntos proporcional al tamaño de la fuente y deberíamos evitar bloques de texto de más de 20 palabras por línea.



Test de autoevaluación

¿Cuál es la unidad más simple de comunicación visual?

- a) La letra
- b) El color
- c) El punto
- d) El cuadrado

¿Qué es una tipografía serif?

- a) Una tipografía de letra ligada con adornos.
- b) Una tipografía formada solo por mayúsculas.
- c) Una tipografía de letra no ligada.
- d) Una tipografía formada solo por minúsculas.

¿Qué no ha de indicar la guía de estilos?

- a) iconografía.
- b) Funcionamiento de formularios.
- c) Espaciado del texto.
- d) Tecnologías utilizadas



Recursos y enlaces

- [Adobe Color CC](#) herramienta online para crear combinaciones de colores.
- [Colrd](#) ejemplos de paletas de colores y texturas.
- [CheckMyClorours](#) herramienta online para testear los colores de una web.
- [Freephotos.cc](#) es una web que ofrece imágenes para usar en la web de forma gratuita.
- [Mystockphotos](#) es una web que ofrece imágenes para usar en la web de forma gratuita.
- [Patternlibrary](#) contiene un conjunto de tramas y texturas para usar de forma gratuita.
- [Iconfinder](#) es una web con miles de iconos gratuitos y de pago.
- Icons8: <https://iconos8.es/music/> . web con iconos, imágenes, músicas con licencia CC
- [Thenounproject](#) es una comunidad de desarrolladores de iconos.
- Guía de estilos de Watsapp: <https://whatsappbrand.com>
- Guía de estilos de Kickstarter : https://www.kickstarter.com/help/brand_assets
- Guía de estilos de Ben&Jerry's: <http://lamp-global.benjerry.com/redesign/style-guide/>
- Guía de estilos de Facebook: <https://en.facebookbrand.com/#brand-guidelines-assets>
- Especificación oficial para crear un SiteMap XML:
<https://www.sitemaps.org/es/protocol.html>

Conceptos clave

- **Guía de estilos:** reglas, interacciones y elementos gráficos que deberemos utilizar en una interfaz.
- **Wireframe, prototipo, mockup:** esquema básico sobre los elementos que van a componer una interfaz.
- **Serif:** letra con decoración. Letra ligada.
- **Sans-serif:** letra sin decoración.



Ponlo en práctica

Actividad 1

Busca un ejemplo de los siguientes sitios web e indica el porqué de sus colores:

- Empresa de transporte.
- Web de comida rápida.
- Web que ofrezca servicios de tecnológicos con computadoras.



SOLUCIÓN

VERSIÓN IMPRIMIBLE ALUMNO LINKIAFP



Tema 2: Definición y aplicación de estilos

¿Qué aprenderás?

- Crear hojas de estilos seleccionando los colores y tipografías adecuadas a la aplicación.
- Maquetar la web siguiendo los prototipos y guía de estilos.

¿Sabías que...?

- El W3C presentó el 5 de mayo de 1999 las primeras pautas de accesibilidad para el contenido web.
- Se calcula que el 19% de la población en Estados Unidos tiene alguna discapacidad que puede dificultar el acceso al contenido web.
- Para que las personas ciegas puedan interactuar con la web los lectores de pantalla convierten el texto en voz sintetizada.



2. Definición y aplicación de estilos

2.1. Qué son las hojas de estilos

Una vez creada la estructura de una web con Html va a ser necesario aplicar estilos para establecer los tamaños y colores de los distintos elementos que conformarán una página web. Si el código Html contiene la información de la web estructurada en etiquetas, las hojas de estilo en cascada van a contener los estilos que se aplicarán a los elementos html.

Cuando hablemos de un elemento html o un elemento html nos referiremos a el conjunto de una etiqueta inicial, su contenido y la etiqueta final.

Comúnmente nos referiremos a las hojas de estilos como archivos CSS (Cascading Style Sheet). Cada hoja de estilo será un archivo de texto plano con extensión “.css” que contendrá un conjunto de reglas para seleccionar distintos elementos html y aplicarles unos estilos. Cada documento html va a poder tener vinculados tantos archivos CSS como sea necesario.

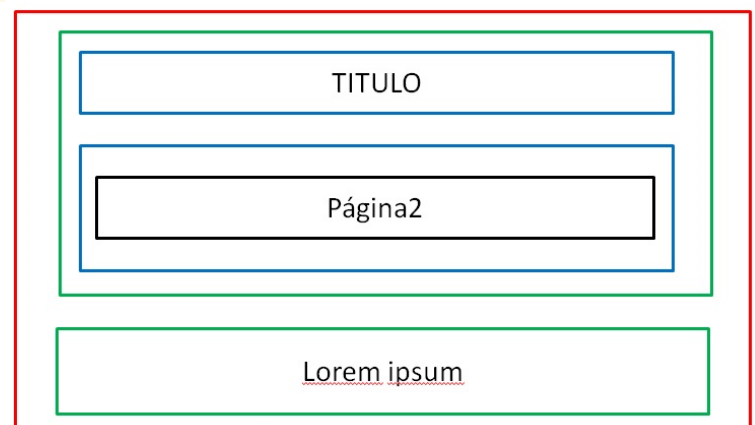
Cuando un navegador interpreta un código html siempre lo lee de arriba hacia abajo y lo muestra por pantalla dibujando los tags de izquierda hacia derecha. Pero, ¿cómo interpreta los elementos un navegador? Vamos a ver el Box Model.

2.1.1. El modelo de cajas

El modelo de cajas (Box Model) especifica que, cuando un navegador tiene que renderizar un código html, debe empieza siempre por la primera línea leyendo de izquierda a derecha interpretando cada etiqueta del body como una caja ubicada en la misma posición jerárquica que el html. Estas cajas son dibujadas por defecto sin ningún borde ni color, ocupando toda la anchura disponible y con una altura igual a su contenido.

En el siguiente ejemplo se muestra una representación de como el navegador interpretaría el siguiente código html según el modelo de cajas. Los colores y espaciados son una referencia para visualizar las cajas generadas.

```
<body>
  <div>
    <h1>TITULO</h1>
    <nav>
      <a href="pagina2.html">
        Página2
      </a>
    </nav>
  </div>
  <div>Lorem ipsum</div>
</body>
```





Representación del modelo de cajas creado a partir de un documento html

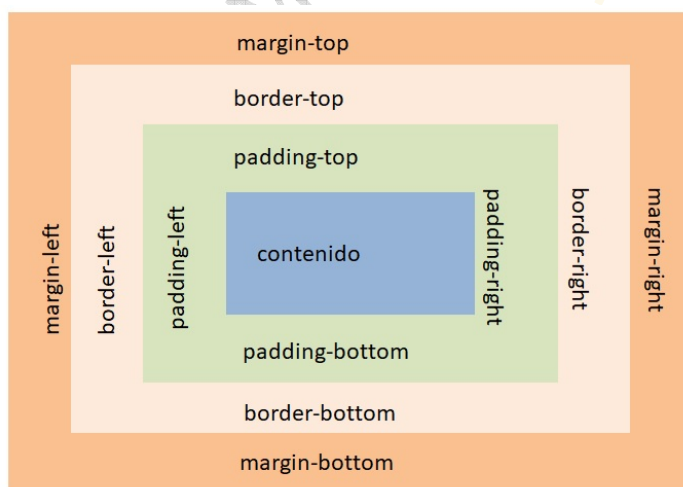
Si dos etiquetas son hermanas se dibujarán una debajo de la otra, si son madre e hija se dibujará la etiqueta hija dentro de la madre.

Las cajas que genera un navegador no tienen ningún color ni tamaño asociados de por sí. Cada caja está compuesta por distintos componentes van a poder tener un conjunto de propiedades para determinar cómo mostrarlos por pantalla. Estos componentes son:

- **margin:** representa el margen exterior entre una caja y el resto. Su única propiedad será la anchura. El margen siempre va a estar situado alrededor del borde de la caja.
- **border:** representa el borde visible de la caja. Tiene como propiedades el color, anchura y estilo. El borde va a estar situado siempre alrededor de la caja.
- **la caja:** propiamente dicha la caja es el espacio contenido inmediatamente dentro del borde. Contiene un gran número de propiedades con las que podremos modificar sus medidas, color de texto, color de fondo, estilos de texto, etc....
- **padding:** representa el espacio entre el borde y el contenido de la caja. Su única propiedad va a ser la anchura y su fondo será el de la propia caja. El padding siempre va a estar entre la caja y su contenido.
- **Contenido:** es el espacio que contiene el texto u otras etiquetas situadas jerárquicamente dentro de la caja actual. No tiene propiedades específicas, pero se verá afectado por todas las propiedades aplicadas a la caja.

Podremos además especificar si queremos modificar el estilo superior, derecho, inferior o izquierdo del margin, border o padding.

En el siguiente ejemplo se muestra una representación de una caja con todos sus componentes:



Componentes de una caja



Ahora que ya conocemos como el navegador interpreta cada elemento y los componentes que forman cada caja, vamos a ver cómo podemos cambiar los estilos de éstas.

2.2. Crear hojas de estilos

Las hojas de estilos van a ser documentos de texto llano que van a contener toda la información para determinar cómo queremos que se muestren los elementos HTML.

A un documento HTML le podemos vincular tantas hojas de estilo CSS como nos sean necesarias.

Una hoja de estilos va a estar formada por un conjunto de reglas CSS. Una regla CSS seleccionará un conjunto de etiquetas y cambiará a los distintos componentes de sus cajas un conjunto de estilos.

Una regla CSS va a estar formada por uno o varios selectores precediendo al conjunto propiedades que se van a modificar agrupadas entre llaves { }. Las propiedades siempre se definirán con la sintaxis: *nombre de la propiedad: valor de la propiedad;*

```
Selector1, selector2 {  
    propiedad1: valor;  
    propiedad2: valor;  
}  
selector3 {  
    propiedad1: valor;  
}
```

Estructura de ejemplo de dos reglas CSS formadas por selectores, propiedades y valores

Para que el navegador sepa qué hoja de estilos tiene que aplicar a un documento HTML es necesario que vinculemos el documento HTML con el CSS.

Para vincular a un documento HTML un archivo CSS utilizaremos la etiqueta <link> especificando en su atributo "href" la ruta del archivo CSS relativa desde el propio documento HTML, en su atributo "rel" especificaremos el valor "stylesheet" y por último en su atributo "type" especificaremos el valor "text/css".

Por ejemplo, para vincular a un documento un archivo CSS de nombre "ejemplo.css" ubicado en la misma carpeta que el documento HTML añadiremos la etiqueta:

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

Se recomienda añadir ésta etiqueta dentro del elemento <head> del HTML.



Desde un documento CSS le podemos vincular otro documento CSS con la sentencia `@import` . En el siguiente ejemplo vemos las dos sintaxis posibles:

```
@import "ejemplo2.css";  
@import url ("ejemplo2.css");
```

No se recomienda utilizar el `@import` porque ralentiza la generación de la web.

También es posible aplicar estilos CSS sin tener que crear una hoja de estilos mediante dos técnicas:

- Añadiendo dentro del mismo documento HTML un tag `<style>` con las reglas que queramos aplicar a éste documento HTML.
- Añadiendo a una etiqueta un atributo `"style"` con el conjunto de propiedades y valores que queremos aplicar a ésta etiqueta en concreto.

```
<style type="text/css">  
div {  
color: red;  
}  
</style>  
<div style="background-color: blue">  
Lorem ipsum  
</div>
```

Definiendo los estilos CSS en el mismo documento HTML

Definir las reglas CSS en el propio documento HTML va en contra del paradigma deseado en el que el documento HTML solo ha de contener la información y su estructura. A demás no permite reaprovechar las reglas definidas para otro documento.

Si en algún momento queremos escribir comentarios en un documento CSS siempre se van a escribir entre la marca `/*` y `*/` . Los comentarios escritos entre éstas dos marcas van a poder ocupar más de una línea y nunca van a ser interpretados como reglas CSS ni interferirán en el documento CSS. Muy importante remarcar que escribir comentarios utilizando la marca `//` es completamente incorrecto e inducirá a errores cuando el navegador interprete las reglas.

Ahora que ya sabemos cómo el navegador interpreta los elementos HTML y cómo podemos crear y vincular hojas de estilos, vamos a ver cómo podemos seleccionar las distintas marcas para cambiar sus propiedades



2.3. Los Selectores

Los selectores CSS nos van a permitir seleccionar un conjunto de elementos para aplicarles estilos. Hay más de cincuenta tipos de selectores distintos, muchos de ellos derivados y otros poco usados, por lo que vamos a centrarnos en aquellos selectores primordiales y ver cómo funciona seleccionando distintos elementos del siguiente código HTML de ejemplo.

```
<body>
  <span> Ejemplo CSS </span>
  <ul>
    <li id="primerLi"> 1 </li>
    <li> 2 </li>
    <li class="destacado"> 3 </li>
    <li> <span>4</span></li>
  </ul>
  <span class="destacado">
    Lorem ipsum
  </span>
</body>
```

Código HTML sobre el que ver los distintos selectores CSS mostrados en la tabla



selector	Ejemplo	Descripción	Elementos seleccionados
Nombre del elemento	span { }	Directamente con el nombre de un elemento selecciona todos los elementos del tipo indicado.	<code>Ejemplo CSS </code> <code>4</code> <code></code> Lorem ipsum
#identificador	#primerLi	Una # seguido de un nombre selecciona aquél elemento con un atributo <i>id</i> igual al indicado El valor de un atributo <i>id</i> nunca se puede repetir en un mismo HTML	<code><li id="primerLi"> 1 </code>
.nombreClase	.destacado	Un punto seguido de un nombre selecciona a todos aquellos elementos con un atributo <i>class</i> igual al seleccionado	<code><li class="destacado"> 3 </code> <code></code> Lorem ipsum
selector.selector	li.destacado	Dos selectores unidos por un punto seleccionan los elementos que cumplan el primer y el segundo selector	<code><li class="destacado"> 3 </code>
selector selector	ul .destacado	Un espacio entre selectores selecciona todos los elementos del selector de la derecha dentro de los elementos seleccionados por el selector de la izquierda	<code><li class="destacado"> 3 </code>
selector>selector	body>span	Un > entre selectores selecciona todos los elementos seleccionados por el selector de la izquierda hijos del selector de la derecha	<code> Ejemplo CSS</code> <code></code> Lorem ipsum



<i>selector,selector</i>	#primerLi,span	Una , entre selectores selecciona todos los elementos del primer selector y del segundo selector	<li id="primerLi"> 1 Ejemplo CSS 4 Lorem ipsum
<i>selector+selector</i>	#primerLi + li	Un + entre selectores selecciona los elementos del segundo selector situados después del primer selector	 2 <li class="destacado"> 3 4
<i>selector~selector</i>	#primerLi ~ li	Un ~entre selectores selecciona el elemento del segundo selector precedido por el primer selector	 2
[nombreAtributo]	[id]	Un nombre de atributo entre [] selecciona los elementos que tengan ese atributo	<li id="primerLi"> 1
<i>selector:hover</i>	li:hover	Un selector seguido de :hover selecciona los elementos del selector cuando el ratón se sitúe encima suyo	Seleccionará los elementos li cuando el ratón se sitúe encima algun elemento li.
<i>selector:nth-child(expression n)</i>	li:nth-child(2)	Un selector seguido de :nth-child() filtra aquellos elementos seleccionados por el selector que además como hijos ocupen una cierta posición establecida por expresión n. Si n es un número, selecciona solo el hijo situado en esa posición.	Selecciona el segundo hijo: 2



<i>selector:nth-child(expression n)</i>	li:nth-child(n+3)	Podemos utilizar una variable n para asignar valores variables. N adquirirá automáticamente valores desde el 0 hasta el infinito.	Selecciona los <i>li</i> a partir del tercer hijo: <li class="destacado"> 3 4
<i>selector:nth-child(expression n)</i>	li:nth-child(2n+1)	A n también podemos multiplicarle un número	Selecciona los <i>li</i> situados en posiciones impares: (2*0+1), (2*1+1), (2*2+1), <li id="primerLi"> 1 <li class="destacado"> 3

Si en algún momento se sobreponen dos reglas CSS aplicando distintos estilos a una misma etiqueta html, se aplicará aquella regla con el selector más específico. En caso que ambas reglas tengan el mismo selector se aplicará aquella regla que se haya definido posteriormente. Se puede aumentar la prioridad de una propiedad añadiendo el modificador **important** como último parámetro de su valor.

En el siguiente ejemplo CSS el elemento li con id #primerLi terminará con un color verde (por el !important) y un color de fondo amarillo, ya que el último selector es igual de específico que el primero.

```
#primerLi{  
  color:red;  
  background-color: blue;  
}  
li{  
  color:green !important;  
  background-color: violet;  
}  
#primerLi{  
  background-color: yellow;  
}
```

Ahora que hemos visto cómo podemos seleccionar distintos elementos vamos a ver que propiedades podemos modificar de sus respectivas cajas.



2.4. Propiedades

Las propiedades CSS siempre se escriben mediante la sintaxis: *nombre-propiedad:valor;*. Algunas propiedades admiten valores compuestos, en éste caso los valores se separarán por espacios hasta terminar con el punto y coma. Antes de ver las propiedades vamos a ver los principales valores que podremos asignar a cada propiedad CSS.

tipo	unidad	ejemplo	Descripción
Longitud y tamaños	px	10px	Píxeles. Indica el número de píxeles que va a ocupar. Es una unidad muy utilizada.
Longitud y tamaños	em rem	3em 3rem	1em y 1rem es el ancho de la letra M en el dispositivo actual. Es un tamaño que va a variar en gran medida según el dispositivo.
Longitud y tamaños	%	50%	Porcentaje. El tamaño será un porcentaje del tamaño del padre. Es una unidad muy utilizada en anchuras y poco recomendable en alturas. Si el padre no tiene un tamaño asignado, él valor puede ser 0 (sobre todo en alturas)
Longitud y tamaños	0	0	Elimina el valor de una propiedad.
General	Auto	auto	Indica el valor por defecto de una propiedad. En altura y anchuras indica un tamaño igual a su contenido. En fuentes indica los valores heredados.
Colores	nombre en inglés	blue	css acepta un largo listado de colores preestablecidos escritos en inglés



Colores	Valor hexadecimal	#ff0000	<p>Los dos primeros dígitos indican la cantidad de Rojo que contendrá, los dos segundos la cantidad de Verde y los dos últimos la cantidad de Azul. Cada dígito es un número hexadecimal desde el 0 hasta el f. El 0 indica a ausencia de un color y la f la saturación máxima de ése color. En vez de 6 dígitos podemos utilizar 3 asignando cada uno a un color.</p> <p>El negro será la ausencia de todo color: #000 y el blanco la composición de todos los colores: #fff. El rojo será #f00, el verde #0f0 y el azul #00f.</p>
Colores	RGB	rgb(255,0,0)	<p>Cada número entre los paréntesis () indicarán con un número del 0 al 255 la cantidad de color Rojo, Verde y Azul, en donde el 0 es la ausencia del color y 255 es la máxima saturación.</p> <p>El negro será el rgb(0,0,0), el blanco el rgb(255,255,255), el rojo rgb(255,0,0) ,el verde el rgb(0,255,0) y el azul el rgb(0,0,255).</p>
Colores	RGBA	rgba(255,0,0,0.5)	<p>Permite indicar un color con una opacidad. La cantidad de opacidad de un color vendrá dado por el último valor y será un número entre 0 y 1, en donde el 0 es totalmente transparente y 1 es completamente opaco.</p>

Tabla de los principales valores para las propiedades CSS



Ahora que hemos visto los valores que podremos asignar vamos a ver las principales propiedades CSS que podremos modificar del modelo de caja de cualquier elemento HTML.

propiedad	ejemplo	Descripción
Font-family	font-family: sans-serif;	El tipo de fuente de la letra. Si el nombre de la fuente contiene espacios se escribe entre comillas.
Font-size	font-size: 1.5rem;	El tamaño de la fuente
font-weight	font-weight:bold;	El ancho de la fuente. Los valores principales son: “normal” sin negrita o “bold” para indicar en negrita.
Font-style	font-style:italic;	Si queremos que la fuente esté en cursiva (con “italic”) o sin cursiva (con “normal”)
Color	color:red;	Establece el color del texto.
Letter-spacing	letter-spacing:0.5em;	Espaciado entre distintas letras. El valor por defecto es “normal”
Line-height	line-height:1em;	Espaciado entre líneas.
Text-align	text-align:justify;	Alinea el texto a la izquierda (“left”) , derecha (“right”), en el centro (“center”) o justificado (“justify”).
Text-decoration	text-decoration: none;	Decora el texto subrayándolo (“underline”) , tachándolo (“line-through”), con una línea por encima (“overline”) o sin decoración (“none”).
Border-width	border-width:2%;	Anchura de un borde
Border-style	border-style:solid;	Estilo del borde. Sus principales valores son: sin borde (none), solid (normal), a puntos (dotted) , a rayas (dashed), doble raya (double).



Border-color	border-color:blue;	El color del borde.
Border	border: 2% solid blue;	La anchura, el estilo y el color de todos los bordes de una caja. Primero se debe indicar la anchura, luego el estilo y por último el color.
Border-top/ border-right / border-bottom / border-left	border-left: 2% solid blue;	La anchura, el estilo y el color del borde superior, derecho, inferior o izquierdo. Primero se debe indicar la anchura, luego el estilo y por último el color.
Border-radius	border-radius: 2px 5px 10px;	La curvatura de cada uno de las esquinas de una caja. El primer valor se corresponde con la superior derecha y sucesivamente los valores van seleccionando las esquinas en el orden del reloj. Un valor de 0 deja la esquina sin curvatura, a mayor valor mayor curvatura.
Margin	margin: 10%;	La anchura de todos los márgenes de la caja.
Margin-top / margin-right / margin-bottom / margin-left	margin-right: 20px;	La anchura de uno de los márgenes superior, derecho, inferior o izquierdo de la caja.
Padding	padding:2em;	La anchura de todos los márgenes internos de la caja.
padding-top / padding-right / padding-bottom / padding-left	margin-right: 20px;	La anchura de uno de los márgenes superior, derecho, inferior o izquierdo de la caja.












Box-shadow	box-shadow: 5px 2px 4px red;	Añade un recuadro del color a modo de sombra por detrás del elemento. El primer parámetro indica su desplazamiento horizontal, el segundo el desplazamiento vertical, el tercero el difuminado de la sombra y el cuarto el color.
background-color	background-color:red;	El color de fondo del elemento
background-image	background-image: url("imagen.jpg");	Una imagen de fondo para el elemento. La ruta a la imagen es relativa a la ubicación de la hoja de estilos.
background-position	background-position:50% 50%;	Donde va a empezar a mostrarse la imagen de fondo. El primer valor indica la posición horizontal y el segundo la posición vertical.
background-repeat	background-repeat: repeat-x;	Indica si queremos que una imagen de fondo se repita en el caso que ocupe menos que el elemento. Podemos indicar que se repita siempre (repeat), solo en el eje vertical (repeat-y), solo en el eje horizontal (repeat-h) , no se repita (no-repeat), se repita añadiendo espacios (space)
background-size	background-size:100% 50px;	La anchura y altura de la imagen de fondo dentro del elemento contenido.
Width	width:50%;	Anchura del elemento
Height	height:20px;	Altura del elemento
Min-width / Max-width	min-width: 200px; max-width: 400px;	Valor mínimo y máximo que puede tener la anchura de un elemento.
Min- height / Max-height	min-height:100px max-height:300px	Valor mínimo y máximo que puede tener la altura de un elemento.



Overflow	overflow:scroll;	Si queremos mostrar las barras de desplazamiento para mostrar todo el contenido de un elemento. Podemos indicar que se muestren siempre (scroll) , nunca y se oculte el contenido que sobresalga (hidden), nunca y se muestre el contenido que sobresalga (visible)
Overflow-x / overflow-y	overflow-x:hidden; overflow-y:auto;	Si queremos mostrar las barras de desplazamiento en el eje de las x o en el eje de las y.
top / right / bottom / left	top:30px; left:10%;	<p>Desplaza un elemento desde su posición natural. El desplazamiento se puede producir o bien desde la posición superior (top), posición derecha (right), posición inferior (bottom) o posición izquierda (left).</p> <p>Si se establece un “top” no se puede establecer un “bottom” y viceversa. Asimismo si se establece un “right” no se puede establecer un “left” y viceversa.</p> <p>El porcentaje hace referencia a la anchura (para right y left) o altura (para top y bottom) del elemento padre escogido como referencia.</p>
z-index	z-index:4;	Indica la prioridad en mostrar un elemento cuando dos elementos se sobreponen. Su valor inicial para todos los elementos es de 0 y puede aceptar cualquier número entero. Si dos elementos tienen el mismo valor se muestra el último elemento definido en el HTML.



Cursor	<code>cursor:pointer;</code>	<p>Tipo de cursor que se va a mostrar cuando el ratón se sitúe encima de éste elemento. Dependerá del navegador y el sistema operativo. Posibles valores: default ,</p> <p>pointer , wait , move , progress </p> <p>, not-allowed , help , all-scroll </p>
@font-face	<pre>@font-face{ font-family:miFuente; src: url(fuente.woff); }</pre>	<p>Define una nueva fuente a partir de un archivo que contenga la fuente. El nombre de la nueva fuente será el valor de "font-family". El parámetro "src" admite archivos del tipo ttf, otf, woff (otros formatos pueden no ser soportados por el navegador)</p>
Position	<code>position:fixed;</code>	<p>Qué elementos se tomarán como referencia para situar un elemento. Posibles valores:</p> <ul style="list-style-type: none"> • Static: valor por defecto. Se muestra según aparezca en el HTML. No admite top, left, bottom o right ni tampoco width ni height. • Absolute: se posiciona según su primer ancestro que no sea static. No tiene en cuenta a sus hermanos. • Fixed: se posiciona según la ventana del navegador. No tiene en cuenta ni a sus hermanos ni a sus ancestros. • Relative: se posiciona según su situación inicial. A diferencia de static acepta top, left, bottom y right. • Sticky: se posiciona como relative excepto cuando el scroll oculta al elemento, entonces se sitúa como en fixed.



display	display:inline;	<p>Como se va a comportar una caja con respecto al resto de cajas. Hay muchos tipos de distintos de display , pero nos centraremos en los principales:</p> <ul style="list-style-type: none">• block: no permite otras cajas a su lado. Por defecto ocupa todo el ancho disponible.• inline: permite otras cajas a su lado. Siempre ocupa el tamaño justo para mostrar su contenido. No acepta width ni height.• Inline-block: permite otras cajas a su lado. Se pueden indicar sus medidas con width y height.• None: no se va a mostrar el elemento ni se va a tener en cuenta para calcular tamaños.
---------	-----------------	--

Ahora que ya sabemos cómo diseñar y aplicar estilos a una página web, vamos a una técnica para codificar-la y pasar de un diseño a un código HTML y CSS, en definitiva vamos a ver cómo podemos maquetar una web.

2.5. Maquetar con display inline-block

Maquetar una web consiste en transformar un diseño o prototipo en un documento HTML y CSS funcional. Hay muchas técnicas distintas de maquetación en base a los distintos tipos de display disponibles. Nos centraremos nosotros en el display inline-block y por ello vamos a conocer más a fondo las propiedades que interactúan con él.

2.5.1. Display inline-block

Un elemento con un display:inline-block admite a otros elementos a su lado y también tamaños.

Utilizando el display:inline-block hay que tener presente que si utilizamos porcentajes para indicar las anchuras de los elementos solo se van a situar uno al lado del otro si sus anchuras suman como máximo 100%. Al calcular éstas anchuras también se tienen en cuenta los caracteres o espacios que puedan haber entre ellos, por ello es preferible eliminar en el HTML cualquier espacio entre los tags que se quieran situar uno al lado del otro.



En el siguiente ejemplo vemos como para un mismo CSS cuando exista una separación entre los elementos la suma de sus anchuras va a exceder del 100% mostrando los elementos uno debajo del otro.

```
body, html{
  width: 100%; height: 100%;
  padding: 0; margin: 0;
}


div{
  position: relative;
  display: inline-block;
  vertical-align: top;
  height: 50px;
  width: 100%;
  top: 0; left: 0;
}

div:nth-child(1){
  width: 50%;
  background-color: #6699ff;
}

div:nth-child(2){
  width: 50%;
  background-color: #66ff66;
}
```

```
<body>
  <div>
    Div1
  </div>
  <div>
    Div2
  </div>
</body>
```

```
<body>
  <div>
    Div1
  </div><div>
    Div2
  </div>
</body>
```




Ejemplo del problema causado por el salto de línea entre elementos

2.5.2. Text-align

Utilizaremos el `display: inline-block` para situar muy fácilmente los elementos de forma centrada, justificada o alineada a la derecha o izquierda utilizando el estilo `text-align`. Hemos de tener presente que `text-align` no alinea el elemento al cual se le aplica el estilo, sino a los elementos que contiene y que sean del tipo `inline` o `inline-block`.

En el siguiente ejemplo utilizamos el `text-align` para alinear el elemento `.centrado` dentro del elemento `.quienCentra`.


Div que centra
Div Centrado

```
<div class="quienCentra">
  Div que centra <br />
  <div class="centrado">
    Div Centrado
  </div>
</div>
```

```
div{
  position: relative;
  display: inline-block;
  vertical-align: top;
  width: 40px; height: 50px;
  top: 0; left: 0;
}

.quienCentra{
  text-align: center;
  height: 200px;
  width: 300px;
  background-color: #6699ff;
}

.centrado{
  width: 100px;
  height: 100px;
  background-color: #ffcc66;
}
```



Ejemplo del uso del `text-align` para centrar un elemento `inline-block` dentro de otro elemento

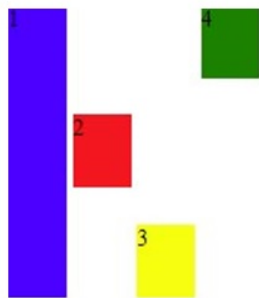


2.5.3. Vertical-align

Nos va servir para indicar la alineación vertical de un elemento inline-block respecto al primer elemento inline-block de la fila. Por defecto tiene el valor “bottom” y normalmente lo querremos con el valor “top”.

En el siguiente ejemplo se muestran 4 divs inline-block con distintos vertical-align.

Al ser inline-block se muestran uno al lado del otro. Para calcular la alineación vertical se toma como referencia el primer div más alto.



```
<body>
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
</body>
```

```
div{ position:relative;
display: inline-block;
vertical-align: top;
width: 40px; height: 50px; top:0; left:0;
}
div:nth-child(1){
height: 200px;
background-color: blue;
vertical-align: middle;
}
div:nth-child(2){
vertical-align: middle;
background-color: red;
}
div:nth-child(3){
vertical-align: bottom;
background-color: yellow;
}
div:nth-child(4){
vertical-align: top;
background-color:green;
}
```

Ejemplo del uso de la propiedad vertical-align para alinear elementos verticalmente

Ahora que hemos visto las principales propiedades que utilizaremos junto a display inline-block vamos a ver qué estrategia utilizaremos para maquetar la web.

2.5.4. Pasos para la maquetación

Empezaremos por indicar que todos aquellos elementos que queramos usar para estructurar la web (DIV, SPAN, ARTICLE, H1, SECTION, NAV, etc.) van a ser del tipo inline-block, con una alineación vertical top, una anchura del 100% , una posición relativa y sin márgenes y paddings. No incluiremos en éste grupo los elementos de tipo TABLE, UL, LI, TD, TR, etc. que tienen su propio tipo de display. Crearemos entonces una regla CSS parecida a la siguiente:



```
div, header, nav, article, aside, footer, p,  
ul, li, h1, h2, h3, h4 {  
    display: inline-block;  
    position: relative;  
    vertical-align: top;  
    width: 100%;  
    top: 0;  
    left: 0;  
    padding: 0;  
    margin: 0;  
}
```

Regla CSS inicial que utilizaremos para maquetar con display inline-block

Estableceremos la anchura del elemento *body* y *html* a 100% y eliminaremos sus márgenes y paddings para asegurar que los porcentajes en anchuras se calculan correctamente y quitar los márgenes o paddings que pueda establecer por defecto el navegador.

```
html, body {  
    width: 100%;  
    margin: 0;  
    padding: 0;  
}
```

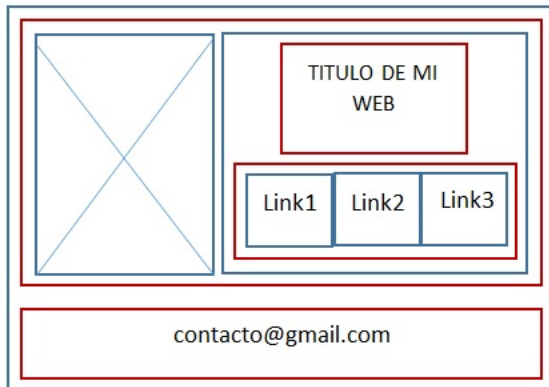
Regla CSS inicial para restablecer el htm y el body

A continuación empezaremos a ir dividiendo el prototipo en cajas horizontales y verticales sucesivamente. Cuando queramos que varias cajas se sitúen una al lado de la otra les tendremos que indicar sus anchuras.

Recuerda que si un elemento no cabe al lado de otro elemento se va a mostrar debajo de él. Recuerda también en eliminar el espacio entre las marcas de cierre de un elemento e inicio de otro para evitar un espacio adicional.

Para hacer las divisiones podemos utilizar tags semánticos o sin significado como los DIV o SPAN.

En el siguiente ejemplo transformaremos el prototipo de la izquierda en un código html tenemos una caja global que se correspondería con el body. Dentro dividida en 2 horizontales (header y footer). El header dividido en 2 verticales (dos div), y la segunda vertical otra vez dividida en 2 cajas horizontales (h1, div)



```
html, body{ width: 100%;  
            margin:0; padding:0;  
}  
div, header, nav, article, aside,  
footer, p, ul, li, h1, h2, h3, h4{  
    display: inline-block;  
    position: relative;  
    vertical-align: top;  
    width: 100%;  
    top: 0; left: 0;  
    padding: 0; margin: 0;  
}  
.anchura30{ width: 30%;}  
.anchura70{ width: 70%;}  
.contenidoCenter{ text-align: center;}  
.contenidoLeft{ text-align: left;}
```

```
<body>  
  <header>  
    <div class="anchura30">  
        
    </div><div class="anchura70"  
      class="contenidoCenter">  
      <h1 class="anchura70">  
        TITULO DE MI WEB  
      </h1>  
      <div class="contenidoLeft">  
        <a class="anchura30"  
          href="link1.html">Link1</a>  
        <a class="anchura30"  
          href="link2.html">Link2</a>  
        <a class="anchura30"  
          href="link3.html">Link3</a>  
      </div>  
    </div>  
  </header>  
  <footer class="contenidoCenter">  
    <a href="mailto:contacto@gmail.com">  
      contacto@gmail.com  
    </a>  
  </footer>  
</body>
```

Ejemplo completo que muestra el uso del display:inline-block para maquetar la web.



UF1_2_5_4_Display_nline_block: Ejemplo de maquetación con display inline-block



Test de autoevaluación

¿Qué *display* nos permite situar elementos unos al lado de otros pero **no** les podemos dar un tamaño?

- a) block
- b) inline
- c) inline-block
- d) none

¿Qué es 1em?

- a) Lo mismo que 1px.
- b) El color negro.
- c) 1/3 de la diagonal de la pantalla.
- d) el ancho de la letra M en el dispositivo actual.



Recursos y enlaces

- Lista de todos los selectores CSS: https://www.w3schools.com/cssref/css_selectors.asp
- Lista de todas las propiedades CSS: <https://www.w3schools.com/cssref/default.asp>
- Lista de todas las unidades CSS: https://www.w3schools.com/cssref/css_units.asp
- Validador de código HTML: <https://validator.w3.org>
- Validador de código CSS: <https://jigsaw.w3.org/css-validator/>

Conceptos clave

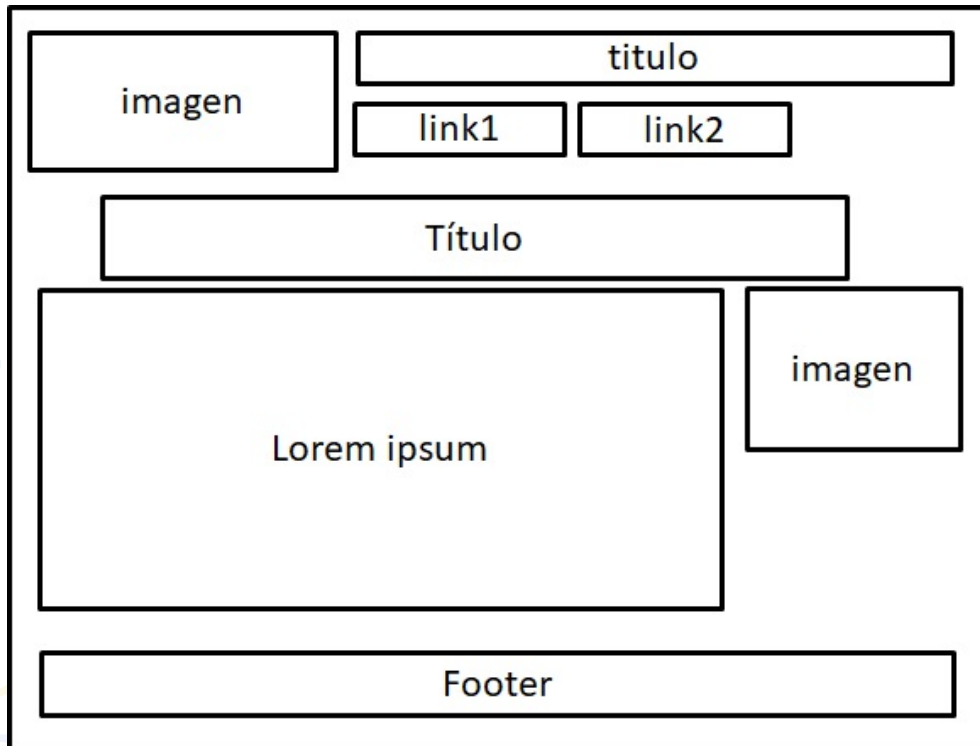
- **BoxModel:** los navegadores muestran por pantalla todos los elementos web como cajas.
- **Elemento:** un elemento html es el conjunto de una etiqueta inicial, su contenido y la etiqueta final.
- **Maquetación web:** proceso en el que transformamos un diseño a un código HTML y CSS.



Ponlo en práctica

Actividad 1

Utiliza display inline-block para maquetar el siguiente wireframe:



SOLUCIÓN