Tema 1 - Planificación de interfaces gráficas

Sitio: Aula Virtual do IES de Teis
Curso: Deseño de interfaces web (DAW-dual 2024-2025)
Libro: Tema 1 - Planificación de interfaces gráficas
Impreso por: Joaquín Lafuente Espino
Data: mércores, 27 de novembro de 2024, 8:15 PM

Táboa de contidos

1. Introducción

Una característica muy común en la mayoría de los sitios web es que tienen una **página principal, home o homepage** desde la que se puede acceder a todos los contenidos ofrecidos por el sitio. Desde el punto de vista del diseño, que es el objetivo de este libro, una máxima es que todas las páginas que componen el sitio web cumplan criterios de homogeneidad y consistencia. Esto se justifica porque un usuario visitante de un sitio web podrá acceder al sitio web a través del home o a través de cualquiera de las páginas web que lo compongan. De alguna manera el diseño web es una **marca comercial o personal**, y esté donde esté el usuario dentro del sitio, las páginas web deben siempre identificarse con esa marca comercial o personal.

Es importante destacar que cuando una persona, empresa o institución crea un sitio web es porque está interesado en **comunicar algo** a los demás, ya sea con fines comerciales o simplemente informativos. Por lo tanto, el objetivo es comunicar, pero con un objetivo específico implícito que se puede resumir en **venderse bien** a los demás. Otra máxima del diseño web es que el sitio web debe ser **atractivo y funcional**. Evidentemente, el diseñador web no puede controlar la información que él o su cliente quiere poner en el sitio, pero sí decide cómo está organizado el sitio y cómo es esa información mostrada.

En este tema identificaremos y describiremos componentes característicos que deben planificarse y diseñarse antes de construir un sitio web. Sin duda, este tema es básico para entender los conceptos y el vocabulario técnico.

2. Elementos de diseño: percepción visual

A la hora de desarrollar un sitio web hay que tener en cuenta qué **interfaz web** ofrecer y qué **elementos** incluir en ella. Cada uno de esos elementos contribuirá a la percepción que sus potenciales usuarios tendrán de la aplicación.

Una de las primeras impresiones que causará la interfaz de usuario será **visual**. En función de la forma, tamaño, ubicación, color, tipografía, etc., que se le asigne a cada uno de los elementos de la interfaz se influirá, de una manera u otra, en el usuario o visitante de un sitio web. El diseñador ha de tener en cuenta constantemente a lo largo de todo su trabajo estas circunstancias y saber valorar la relación directa que puede identificarse entre sus diseños y cómo estos serán percibidos.

El diseñador debe buscar un **equilibrio entre los elementos** que constituyen la interfaz. Debemos tener muy presente que, por principio, nada debe ser totalmente superfluo en un diseño, aunque tampoco es conveniente excederse en la utilización de elementos por el mero hecho de ponerlos, ya que esto puede producir un excesivo ruido o distracciones que pueden enmascarar el mensaje de la comunicación. Una vez asimilada toda la información sobre aquello que quiere comunicar, el diseñador ha de empezar a generar **soluciones de diseño** adecuadas al propósito. Lo primero que determinará es el **área de diseño**, es decir, qué tamaño se asignará al espacio del que se dispone para la composición gráfica. Una composición gráfica puede estar formada por muchos o pocos elementos. Puede componerse exclusivamente de la presencia de texto o solo de imágenes; puede poseer grandes espacios en blanco o constituir una combinación equilibrada de elementos gráficos. Pero, en cualquier caso, debe ser adecuada con lo que se quiere comunicar.

3. Fundamentos de la composición

Aunque no hay ninguna norma específica que garantice el éxito de una composición, sí existen una serie de **pautas** a las que el diseñador se puede adecuar para obtener soluciones eficaces, todas ellas muy relacionadas con la percepción. El diseñador ha de tener un profundo conocimiento de los **factores que rigen el fenómeno de la percepción** para poder establecer sus composiciones de un modo sólido y fundamentado. Algunos de estos factores son:

· **Componentes psicosomáticos del sistema nervioso**: nos facilitan el contacto visual con nuestro mensaje gráfico haciendo uso del mecanismo de percepción llamado vista. Con ella recogemos información visual (percibimos distintas formas, ubicaciones, longitudes de onda de un color, etc.) que luego nuestro cerebro interpreta como contornos, texturas, dimensiones, etc., dotándolas de un significado gráfico definido.

· **Componentes de tipo cultural:** influyen en la interpretación que hacemos de los estímulos desde un punto de vista cultural y educacional. Por ejemplo, el color que en Occidente está relacionado con el luto es el negro mientras que en los países orientales este mismo significado se le asigna al color blanco.

· **Experiencias compartidas con el entorno**: por ejemplo, conceptos altamente arraigados como: hierba/ verde, azul/cielo, hielo/frío. Todas ellas van constituyendo una serie de dualidades que el hombre va aprendiendo desde su infancia y que, posteriormente, serán utilizadas por él como patrones con los que interpretar y dotar de significado la realidad.

Sin embargo, hay más factores relacionados con la disposición de los elementos para conseguir una composición adecuada.

3.1. El equilibrio visual

Antes de hablar del equilibrio visual es necesario definir dos conceptos previos de mucha importancia: el **equilibrio formal** y el **informal**.

El equilibrio formal se basa en la **bisimetría**. Se busca con él un **centro óptico** dentro del diseño y no tiene por qué coincidir con el **centro geométrico** de la composición. El punto de equilibrio formal suele estar ubicado un poco por encima del centro geométrico. Una composición que decida seguir este esquema compositivo reflejará estabilidad, calma y estatismo. No supone una composición muy audaz o creatividad, aunque lo que sí asegura es una distribución armónica de los elementos.

El equilibrio informal, por el contrario, está altamente cargado de fuerza gráfica y dinamismo. Prescinde por completo de la simetría y el equilibrio se consigue aquí en base a contraponer y contrastar los **pesos visuales** de los elementos, buscando diferentes densidades, tanto formales como de color, que consigan armonizar visualmente dentro de una asimetría intencionada.

Las formas pequeñas poseen menor peso visual que las más grandes. Si, además, la forma de la figura no es regular, su peso aumenta notablemente. Los **colores** también juegan un papel importante en lo que respeta al peso visual: cuanto más luminosos sean, mayor peso compositivo tendrán. Entre elementos con el mismo **tamaño**, pero colores de diferente intensidad, tiene más peso visual el de color intenso. Pero, entre elementos del mismo color, tiene más peso visual el de más tamaño.

El último elemento importante de equilibrio es la **posición**. Dependiendo de dónde se coloquen los elementos se podrá conseguir un mayor equilibrio y se apreciarán mejor por parte del usuario. Es evidente (en Occidente), por ejemplo, que los elementos situados en la parte superior de la página tienen más protagonismo que los situados en la parte inferior derecha.

En resumen, para conseguir un equilibrio adecuado hay que estar al tanto de todos los factores compositivos que intervienen, tales como el peso, el tamaño y la posición.

3.2. La tensión compositiva

Lo opuesto al equilibrio desde el punto de vista estructural es la tensión compositiva. La tensión tiene como finalidad dirigir la mirada y conseguir fijar la atención del observador. La tensión se puede conseguir con la combinación de líneas y formas agudas e irregulares. Hay algunas técnicas para provocar la tensión y conseguir captar la atención del usuario. Las principales técnicas son:

Técnica sugestiva: consiste en dirigir intencionadamente la atención a un punto determinado utilizando elementos de apoyo. Por ejemplo, imágenes de personas que miran hacia un punto determinado (que sería el punto de interés).

Técnica rítmica: basada en la tendencia innata del ojo humano a completar secuencias de elementos (ya sean números, formas, figuras geométricas o colores), agrupando aquellos que poseen formas semejantes.

4. Color, tipografía e iconos

Dentro de las composiciones para diseñar sitios web, los elementos más destacados que podemos encontrar en todas ellas son: **colores**, **tipografia** e **iconos**. Los tipos y cantidad de estos elementos, así como su variación, dependerá lo que pretendemos comunicar con el sitio web (y de la creatividad del diseñador). A continuación, se describen estos tres elementos.

4.1. Color

En los entornos gráficos digitales, los colores se forman a partir de tres básicos, el **rojo, verde y azul**, que se denominan **componentes RGB** (del inglés Red, Green, Blue). Generalmente, la **intensidad** de cada componente se expresa (en diseño web) como un número **hexadecimal** del 00 al FF (del 0 al 255 en base diez). Por ejemplo, el color rojo se representa como #FF0000, porque tiene toda la intensidad de rojo y nada de verde y azul.

Los colores básicos son:

#FF0000 - Rojo

#00FF00 - Verde

#0000FF - Azul

Otros colores son:

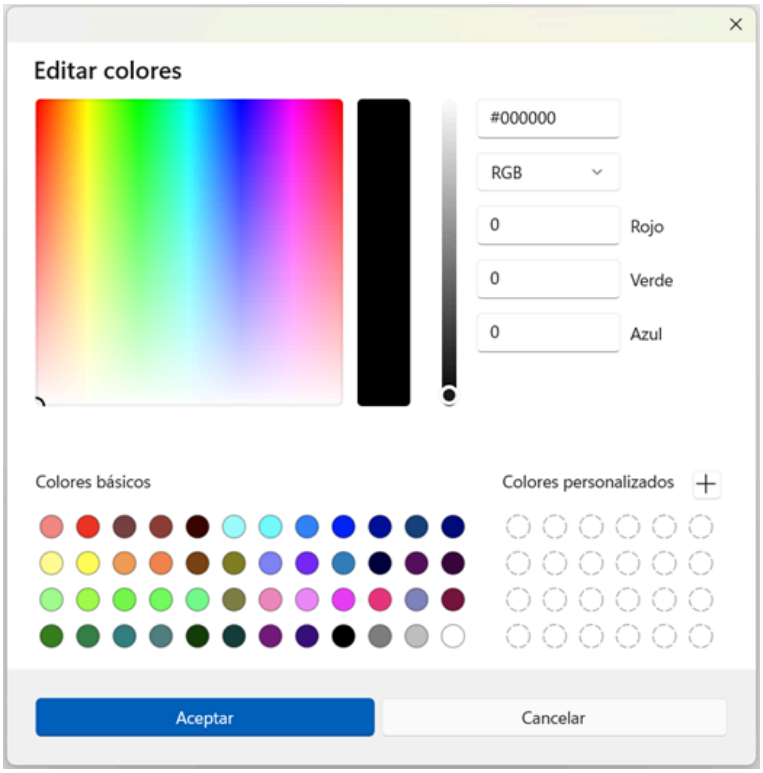
#FFFFFF - Blanco

#000000 - Negro

#FFFF00 - Amarillo (mezcla de rojo y verde)

Para hacer un color más oscuro se reduce la intensidad del componente, dejando los otros dos iguales. De esta forma, el rojo (#FF0000) se hace más oscuro así: #CC0000, #990000, #660000, #330000, etc.

Actualmente, la gran mayoría de entornos que permiten el trabajo con colores ofrecen la equivalencia de los colores en este formato hexadecimal. A continuación, se muestra una típica **paleta de colores** en Windows 11, mostrando el color negro (00 rojo, 00 verde y 00 azul).



Aunque parezca lo más sencillo, elegir una **combinación de colores apropiada** para un diseño es una de las tareas más difíciles. Para algunos expertos en diseño, la combinación adecuada de colores requiere un gen artístico que no todo el mundo tiene. Para otros, la combinación adecuada de colores se puede calcular con ecuaciones matemáticas que combinan colores, tonos y saturación para crear composiciones artísticas.

Sin embargo, para que un diseñador web no tenga que ir de un extremo a otro para poder crear una buena composición, existe software que ayuda en esta labor de crear combinaciones armónicas y placenteras. Estas herramientas suelen estar muy orientadas a facilitar el trabajo del diseñador rescatando combinaciones de sitios o imágenes ya creadas:

· Crear varias combinaciones de colores a partir de un color de referencia.

· Obtener combinaciones de colores presentes en una imagen ya creada. Útil cuando se quiere sacar una combinación, por ejemplo, de una foto.

· Obtener el código de un color y el valor hexadecimal (RGB y CMYK) de cualquier color que se ve en tu pantalla. Esto es interesante cuando se desean sacar los colores de, por ejemplo, un sitio web ya creado.

· Buscar imágenes que satisfagan un patrón de colores concreto. Útil cuando se quiere encontrar imágenes que combinen con los colores de la web.

Algunas de las herramientas actuales son:

· **ColorPix:** es un pequeño software que descargas en tu ordenador y te permite conocer los códigos, las coordenadas y el número de píxeles de cada color presente en tu pantalla. ColorPix traduce automáticamente cada color en los códigos RGB, HEX, HSB y CMYK. Permite, incluso, que hagas zoom a un área de la pantalla. Funciona hasta Windows 10 y es gratuito.

https://web.archive.org/web/20170621044818/http://www.colorschemer.com/colorpix_info.php

· **Instant Eyedropper:** es una herramienta gratuita de detección de color para webmasters que identificará y pegará automáticamente en el portapapeles el código de color HTML de cualquier píxel de la pantalla con sólo un clic del ratón. Compatible hasta Windows 11. Disponible en versión portable e instalable. http://instant-eyedropper.com/

· **Color Schemer Online:** es una aplicación web gratuita útil para crear las mejores combinaciones de colores posibles. Basta con seleccionar los valores RGB o HEX del color con el cual quieres comenzar. Entonces, aparecerán esquemas de colores y gradientes de lo más armoniosos (obtenidos según ecuaciones matemáticas) con los códigos HEX y RGB relativos. https://colorschemer.com/

· **TinEye Color extraction:** es un servicio web gratuito útil para encontrar los colores complementarios para una imagen. Basta con subir una foto al sitio (upload) o encontrar una en la web. Para esa imagen aparecerá una combinación compuesta de los colores primarios de la imagen y su proporción. Sobre esos colores la aplicación propondrá los diez mejores colores únicos, entre los complementarios y los dominantes de la imagen. https://labs.tineye.com/color/

4.2. Tipografía

Sin duda, los **textos** son la base de la gran mayoría de los sitios web. Transmitir información mediante letras es lo más común y, por tanto, requiere una especial atención.

Cuando se habla en diseño web de **fuente** se hace referencia a un conjunto de caracteres con un estilo o modelo gráfico particular. De alguna manera, una fuente es sinónimo de tipo de letra. A la hora de manejar fuentes en un sitio web hay que tener en cuenta una serie de limitaciones y características que complican el diseño. Entre esas limitaciones, la más destacada es que las fuentes disponibles en cada sistema operativo son diferentes. Aunque las versiones actuales de los navegadores instalan un conjunto de fuentes similar en Windows, Linux, MacOs, etc., hay que tener en cuenta que existen otros navegadores y otros sistemas operativos, por lo que es importante asegurarnos de que los contenidos textuales tendrán el mismo aspecto con independencia del navegador que interprete el sitio web. Más allá de esto, las otras limitaciones están relacionadas con la **adecuación**, con lo que se quiere comunicar su legibilidad y, como ocurre con los colores, si son o no combinadas con buen gusto.

Las fuentes más comunes suelen ser las llamadas **Sans Serif**, destacando entre ellas **Verdana, Arial y Helvetica**, aunque hay una fuente concreta con ese nombre, Sans Serif, que hace referencia a un tipo genérico. Estas fuentes son **adecuadas para mostrar texto en pantallas**. Si se desea que los textos se puedan imprimir, es conveniente sustituir las fuentes anteriores por alguna tipo **Serif** (con remates en sus extremos), ya que son **más legibles en documentos impresos** y menos monótonas. Entre estas fuentes tipo Serif destacan las conocidas **Times New Roman, Courier y Courier New**, aunque también hay una fuente concreta llamada Serif que hace referencia a un tipo genérico. Como se verá más adelante, usando CSS es posible indicar que para un mismo texto se usen fuentes diferentes, una para ver en pantalla y otra para que se muestre impresa.

Para concluir, un sitio web no es aconsejable que use más de **tres fuentes**. Es una recomendación bastante extendida en los sitios web actuales. Algunos ejemplos de fuentes se muestran en la siguiente figura:

4.3. Iconos

La palabra **icono** se utiliza para designar a las imágenes gráficas generalmente pequeñas y que suelen ser metáforas de las acciones que se pueden hacer. Por lo general, se trata de mantener una relación entre el icono y lo que representa, es decir, que lo que se identifica con dicho icono está ligado de alguna manera al icono que lo está representando. Respecto a esto, existen algunos estándares de facto, como por ejemplo, el icono de un disquete sustituye a la orden "guardar"; el de una lupa, a la orden "buscar" y el de una carpeta representa a los archivos.

Con estos dibujos **evitamos leer textos** y obtenemos de una manera más rápida las opciones que nos presentan. Una buena elección de estos iconos es muy importante, puesto que si un usuario no es capaz de determinar su significado no hemos conseguido nuestro propósito de ahorrarle tiempo en la visualización de la página. Un icono debe contener la **menor cantidad de detalle posible**, únicamente dejar los imprescindibles para la comprensión de su significado.

Otro punto importante en la elección de un icono es la **estandarización**, o mejor dicho, a lo que están acostumbrados los usuarios. Es muy arriesgado innovar con estos temas puesto que los usuarios son muy reticentes a los cambios y tendría que ser muy bueno el icono para que no despiste al usuario.

La siguiente figura muestra parte del home del sitio web Amazon, un ejemplo de portal que no usa muchos iconos pero sí usa un icono asociado a la cesta de la compra para conseguir más impacto visual, ayudando al usuario a localizar bien cómo acceder a comprar.

Aunque pueda parecer lo contrario, los iconos tienen sus **limitaciones** en la web. Estas limitaciones están relacionadas con la falsa creencia de que un icono es interpretado más rápido por un usuario que un texto. A veces, eso no es así y los iconos, lejos de facilitar, complican más la web, al no estar el usuario familiarizado con lo que quiere representar el icono. Esto es debido a que los iconos son siempre **subjetivos**, están sujetos a la interpretación individual y subjetiva de cada persona a partir de su experiencia. Nunca son totalmente claros e inequívocos y existe riesgo de malentenderlos. Por esta razón no se recomienda usar iconos para **operaciones críticas** y se recomienda mejor un texto con una fuente adecuada y legible, o una combinación de ambos, como la imagen anterior de Amazon.

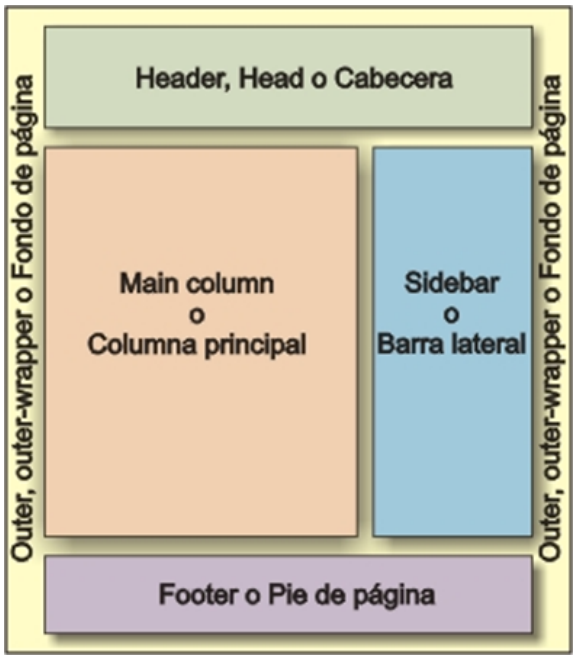
Según varios estudios, debido a los **riesgos de interpretación** de los iconos, su adecuado diseño no puede depender únicamente de la inspiración o preferencias de diseñadores o los responsables del sitio. Es necesaria la creación de varios diseños o prototipos para cada icono y la realización de tests con usuarios reales en un **proceso iterativo de diseño-test-rediseño**.

5. Componentes de una interfaz

Un punto muy importante a la hora de introducir lo que es el diseño de interfaces web es definir cuáles son actualmente sus **componentes**, es decir, qué partes forman un sitio web.

Desde las primeras páginas web hasta la actualidad, los diseños han evolucionado hacia la homogeneidad, ofreciendo unas interfaces bien definidas, con un conjunto de componentes gráficos y funcionales similares que hacen posible que, sea cual sea el usuario que accede a un sitio web cualquiera, la comunicación entre ellos sea posible y efectiva. En esta evolución, como si de una selección natural se tratase, se han asentado **elementos que han demostrado su utilidad** y su comprensión por los usuarios. Algunos ejemplos son: sistemas de navegación, los pies de página o los formularios de entrada de datos, etc., que normalmente encontraremos en todas las páginas web y cuyo diseño y funcionalidad son similares en todas ellas.

La siguiente figura muestra la **estructura general** de un sitio web marcando los nombres de sus elementos principales. A continuación, se describen esos elementos.



5.1. Cabecera

Se entiende por cabecera una zona de la interfaz web situada en la **parte superior** de la misma, de anchura generalmente igual a la de la página y altura variable, en la que se ubica generalmente el logotipo del sitio web o de la empresa propietaria, acompañado generalmente de un texto identificador de la misma y de otros elementos de diseño, como fotografías (simples o formando un montaje), formularios de login (entrada de claves de acceso al sistema), banners publicitarios, etc.

Un ejemplo de cabecera para el sitio web del IES de Teis es mostrado en la siguiente figura. Observar que aparece un menú en la parte inferior y un campo de búsqueda a la derecha.

El **objetivo principal** de la cabecera está muy relacionado con el de las cabeceras en las portadas de la prensa escrita, por ejemplo, diarios:

· Identificar el sitio web con la empresa a la que representa mediante el **logotipo y el nombre** del mismo, de la empresa propietaria o de la marca que representa.

· Identificar y homogeneizar todas las páginas pertenecientes al sitio web, ya que la cabecera suele ser común en todas ellas, creando con ello un **elemento de referencia común**.

· Crear una **separación visual** entre el borde superior de la interfaz y el contenido central de la misma, haciendo más cómoda su visualización y lectura.

El motivo por el que la cabecera se encuentra situada en la zona superior de la interfaz y el logotipo en su parte izquierda obedece a consideraciones de jerarquía visual. En la cultura occidental estamos acostumbrados a leer de arriba hacia abajo y de izquierda a derecha, por lo que la parte superior izquierda de una página es la primera a la que dirige el usuario la vista, con lo que situando en ella el logotipo nos aseguramos que sea el **primer elemento gráfico** que el espectador observe.

La cabecera **no es obligada** en un sitio web, pero es habitual usarla. La forma más común de la cabecera es rectangular, pero conforme avanzan el diseño gráfico, se pueden encontrar de muchas formas y colores, asociándose generalmente al impacto que se quiere causar en el usuario. En cualquier caso, el diseño (colores, tipografia, etc.) de la cabecera nunca debe ocultar el logo y el texto que se muestra en ella.

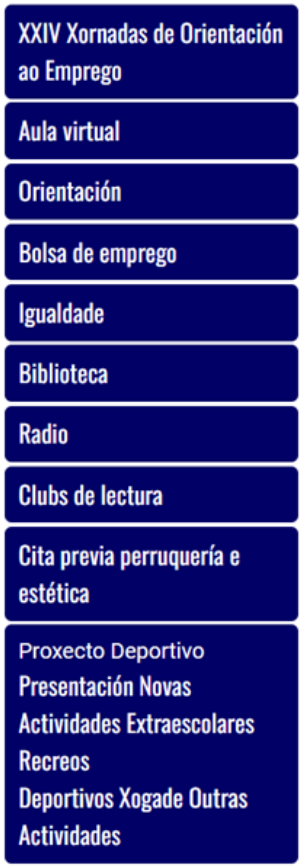
La cabecera no tiene siempre que ocupar todo el ancho de la página, puede ocurrir que tan solo ocupe una parte del mismo, generalmente la izquierda, en la que se suele situar en una banda vertical común con un menú de navegación. Por otro lado, también es posible encontrar **páginas sin cabecera**, generalmente en páginas de inicio que sirven como presentación del sitio y que presentan un diseño especial, diferente al del resto de páginas que lo forman. También ocurre en páginas de diseño vanguardista, que intentan huir de los patrones clásicos. En estos casos el logotipo puede estar situado en cualquier zona de la interfaz, generalmente en la parte inferior izquierda de la misma.

5.2. Los sistemas de navegación

Los sistemas de navegación son los elementos de una interfaz que permiten **moverse por las diferentes secciones** y páginas que componen el sitio web.

Generalmente se presentan como **menús** formados por diferentes **opciones**, con las que el usuario puede interaccionar al seleccionarlas, pasando a una nueva página o documento.

Un ejemplo de menú es mostrado en la siguiente imagen:



Los menús pueden tener textos, gráficos o ambos, todo ello combinado también con efectos dinámicos para acentuar el carácter interactivo de las mismas. Un tipo de efecto es el **rollover**, en el que todos los componentes, una opción o algunos de ellos cambian de aspecto al situar el usuario el puntero sobre ella.

Con capas, CSS y JavaScript es posible crear también menús dinámicos en los que aparecen y desaparecen porciones del mismo según las acciones que haga el usuario sobre sus opciones principales. De este tipo son los conocidos **menús de árbol**, similares al que ofrece el Explorador de Windows para navegar entre los discos duros y sus carpetas, y los **menús de cortinillas**, en los que aparecen y desaparecen capas con grupos de opciones. Un ejemplo de menú de árbol es mostrado en la siguiente figura. La opción “Cómo estudiar” tiene la descendiente “FP Dual” que a su vez tiene varios descendientes más.

Otro tipo de menú muy aceptado es el de **pestañas**, que simula el aspecto de un clásico archivador de carpetas, apareciendo en primer plano la pestaña activa, en un color diferente y unido visualmente a la base común o al cuerpo de la página. Un ejemplo de este tipo de menú es mostrado en la siguiente figura:

Un formato de menú muy extendido es el "**estás aquí**". Este tipo de enlace presenta en forma textual una serie de enlaces que describen la ruta que ha seguido el usuario para llegar a la página actual a partir de la home o página de inicio, permitiendo regresar a cualquiera de ellas rápidamente. Estos menús poseen la ventaja adicional de ubicar al visitante en el total del sitio, con lo que éste sabe en cada momento dónde se encuentra y cómo ha llegado allí. La figura anterior muestra también este formato.

Los menús son un elemento principal en todo sitio web porque permite que el usuario sepa en todo momento cómo moverse por el sitio y saber también dónde está. Por lo tanto, la **ubicación de los menús** es un aspecto muy importante en el diseño. Ésta debe permitir un cómodo acceso a las opciones que lo forman, pero sin llegar a estorbar al resto de elementos.

Los menús tipo lista y los de árbol se sitúan generalmente en la zona lateral izquierda de la página, mientras que el tipo pestaña o "estás aquí" es más habitual verlos en la parte superior, debajo de la cabecera. Esta distribución se ha convertido en un estándar de facto entre los diseñadores, pero el origen de esta costumbre está más ligado a cuestiones técnicas (muy asociada a la resolución y a HTML) que a motivos de usabilidad, funcionalidad o estética.

Si la altura de la página es tal que el usuario tiene que utilizar la barra de desplazamiento vertical tanto que pierde de vista el menú, es conveniente situar una versión reducida del menú principal en el pie de página, para que pueda acceder directamente desde esa posición a las partes del sitio.

Si el menú ofrece un número excesivo de opciones (cinco o más), es aconsejable utilizar menús dobles o menús en forma de árbol que jerarquice las opciones. Esto permitirá que el usuario encuentre la opción deseada con mayor facilidad. En caso de ser necesario, el segundo menú (menú secundario) deberá diseñarse de forma que se identifique claramente como tal, siendo habitual mantener el menú principal como elemento general de navegación del sitio web completo y utilizar el menú secundario para permitir la navegación entre las diferentes páginas de una sección o nivel concreto.

Ejemplo de un sistema doble muy común es el formado por un menú principal lateral y uno secundario ubicado en la zona superior del cuerpo principal de la página, que puede ser de tipo "estás aquí". Otra modalidad común es la formada por un menú principal horizontal bajo la cabecera y uno secundario en el lateral, aunque es posible cualquier combinación lógica y funcional.

5.3. El cuerpo de la página

El cuerpo es la parte de la página web donde se presenta al usuario toda la información referente a los **contenidos de la página**. Lo que aparece en el cuerpo suele ser el objetivo del sitio, lo que el usuario quiere ver. Por lo tanto, el espacio destinado a ella debe ser el mayor de todos, ocupando generalmente entre el 50% y el 85% del total. Su ubicación es siempre central, bajo la cabecera (si la hay) y al lado del menú lateral de navegación (si lo hay).

Los contenidos específicos del cuerpo de la página variarán según sea una página textual, un formulario, una ficha, una tabla o una página mixta, pero aparte de estas particularidades, existirán algunos elementos característicos de esta zona, que deberán estar presentes generalmente en todos los casos.

Es habitual que el cuerpo central lleve un **título** que identifique claramente la página a la que ha accedido el usuario. Este título se situará en la parte superior de esta zona y puede ser reforzado mediante un menú de navegación tipo "estás aquí". El tamaño de las letras del título de página debe ser superior al del resto de los contenidos (como ocurre en los periódicos), con la finalidad de resaltar. Sin embargo, ésta no es la única manera de resaltar el título con respecto al resto del contenido. Otra alternativa es cambiar el color del título con respecto al contenido. Si el contraste en significativo entre ambos colores, se consigue también resaltarlo.

Es importante que todos los elementos gráficos que situemos dentro del cuerpo de página presenten un aspecto similar al del resto de elementos de la interfaz, respetando el estilo de todo el sitio. La siguiente imagen muestra parte de una página en la que el título se resalta con colores y tamaño distintos del contenido y donde el contenido tiene el mismo diseño que el resto de partes de la página (cabecera y menú).

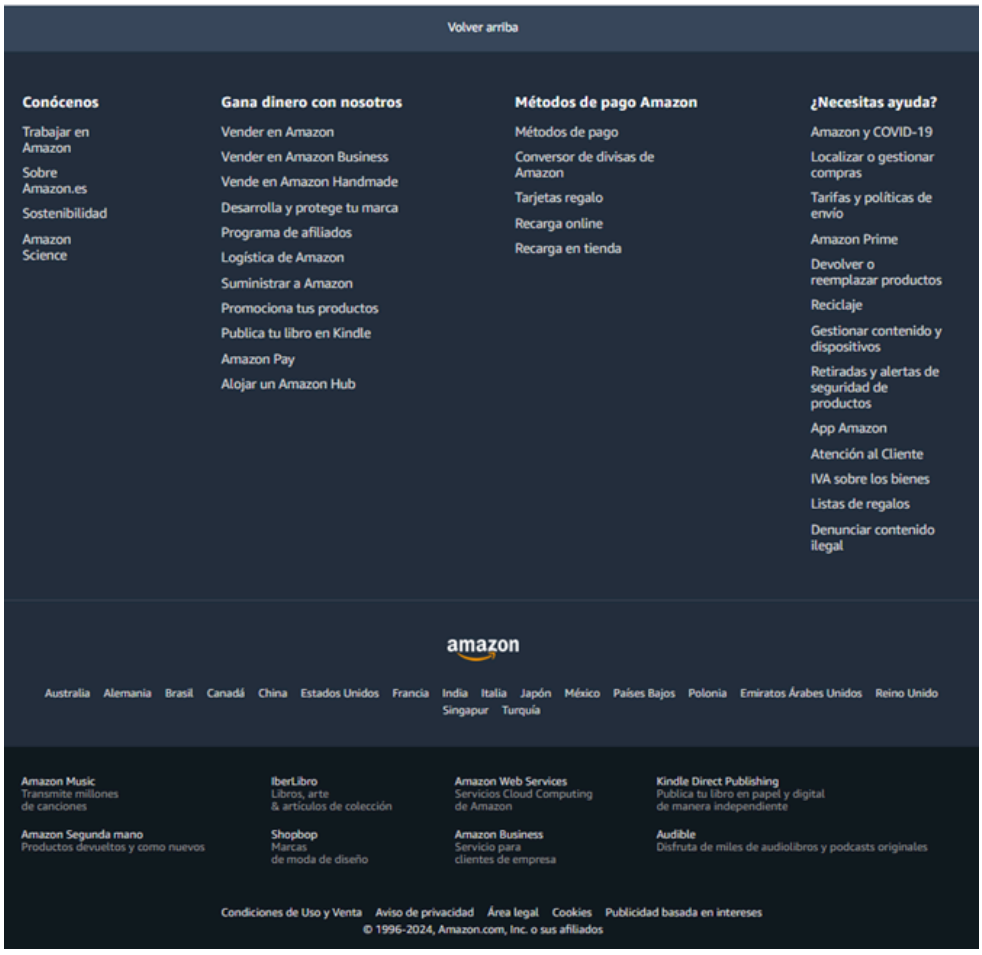
5.4. El pie de página

El pie de página es la parte de una interfaz web situada en la **parte inferior** de la misma, bajo el cuerpo de página. En principio no parece tener una misión muy importante, sin embargo, tiene mucha utilidad por la información que muestra y por ayudar a una **percepción más estructurada** del sitio.

Un uso muy común del pie de página es para mostrar enlaces a **servicios muy particulares** del sitio web, como contratación de publicidad, formulario de contacto, ofertas de empleo, condiciones de uso, políticas de seguridad, etc. Otro uso común es para mostrar **información sobre la empresa** propietaria del sitio web o de su responsable directo.

Como se comentó al hablar de los menús, si la página necesita de mucho movimiento vertical para poder visualizarse entera (usando barra de desplazamiento) el pie de página suele contener un menú auxiliar que permita al usuario continuar navegando por el sitio web sin tener que volver a buscar el menú principal.

Los contenidos del pie de página pueden aparecer alineados de cualquiera de las formas aceptadas (a la izquierda, centrados, a la derecha o justificados), aunque lo normal es que aparezcan centrados en pantalla. Si aparecen alineados de otra manera, siempre deberá estar en consonancia con el resto de elementos de la página. La siguiente figura muestra como ejemplo de pie de página el del portal amazon.es.



5.5. Los espacios en blanco

Aunque parezca mentira, un elemento de especial importancia en un diseño web son los espacios en blanco. Los espacios en blanco se definen como todas esas zonas de la interfaz en las que no hay **ningún otro elemento gráfico**. Entre sus objetivos está el **compensar el peso visual** del resto de elementos, crean márgenes o separaciones entre ellos, encuadrándolos de forma adecuada, y marcan los límites que estructuran la composición, haciendo la interfaz más equilibrada, limpia y bella.

Para muchos expertos en diseño web, la forma correcta es diseñar considerando desde el principio a los espacios en blanco como un **elemento gráfico** más, concibiendo su presencia y su ubicación desde el principio. Los espacios en blanco establecen el lugar, la rejilla base de la composición, que delimita las zonas en las que vamos a situar el resto de elementos, los márgenes y **separaciones** que van a existir entre ellas. A continuación se muestran algunas consideraciones concretas sobre los espacios en blanco.

Si existe un menú lateral de navegación es conveniente dejar siempre un espacio blanco o libre entre éste y el cuerpo de la página. Habrá que dejar, al menos, el mismo espacio entre la cabecera y el cuerpo de página. Si no existe cabecera, la separación será entre el cuerpo y el borde superior de la ventana útil del navegador. Si hemos diseñado una página con dos menús laterales, uno a cada lado, la separación entre estos y el cuerpo de la página será la misma en ambos casos, así como la separación entre los dos menús y los bordes de la ventana. De la misma manera, deberá existir un espacio en blanco de margen entre el dintel o el menú superior y el cuerpo de la página, así como entre éste y el pie de página, que deben tender a ser del mismo alto, buscando la simetría en la composición.

Todas estas separaciones son necesarias para conseguir un diseño poco sobrecargado en el que se delimitan bien las partes de la página.

6. Lenguajes de marcas

Una vez se han visto los elementos principales de un sitio web, es necesario introducir los **lenguajes de creación de la web**. De alguna manera, la web es un programa que se ejecuta en el ordenador y que se define mediante lenguajes de programación que entiendan los navegadores. Por sus características, estos lenguajes son de un tipo especial y reciben el nombre de **lenguajes de marcas. HTML, XML o RDF** son algunos de esos lenguajes.

Un lenguaje de marcado o lenguaje de marcas es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la **estructura del texto o su presentación**. Es decir, un lenguaje de marcas es un tipo de lenguaje que combina texto con información extra acerca del texto. Esa información extra se entremezcla con el texto.

Los lenguajes de marcado son un tipo de lenguajes de programación. Sin embargo, no son exactamente lo mismo, ya que el lenguaje de marcado no tiene funciones aritméticas o variables, como sí poseen los lenguajes de programación.

El lenguaje de marcas más conocido en la actualidad es el **HTML (HyperText Markup Language, Lenguaje de marcado de hipertexto)**, que se utiliza en las páginas web y fue propuesto por Tim Berners Lee en 1989, considerado el padre de la Web.

El lenguaje HTML (Hyper Text Markup Language) es lo que nos permite **describir los contenidos** de una página web de forma textual y estructurada; es un lenguaje para que el navegador conectado a Internet sepa cómo visualizar una página web; es decir, para que se pueda saber si un texto es un título, si éste está centrado o si tiene un tamaña determinado, también permite definir vínculos o enlaces a otras páginas o documentos y gestionar imágenes.

La primera versión de HTML solo mostraba texto con estilo, es decir, contenía etiquetas para títulos, párrafos, listas y viñetas, entre otras características. Muchas empresas utilizaron esta versión de lenguaje como punto de partida, pero incorporando sus propias interpretaciones de las etiquetas. Para organizar el crecimiento de HTML y de la propia Web se creó el **World Wide Web Consortium (W3C)**, que se encargó desde entonces de estandarizar todos los temas relacionados con la Web. Así, en 1995 se publicó la versión 2.0 de HTML.

El lenguaje **HTML 2.0** aportaba compatibilidad con los navegadores y etiquetas adicionales con las que incorporar imágenes, vínculos, tablas y formularios. Con estos últimos se lograba incorporar facilidades de interactividad a nivel de intercambio de información entre el navegador, utilizado por parte del usuario, y un servidor suministrador de las páginas web.

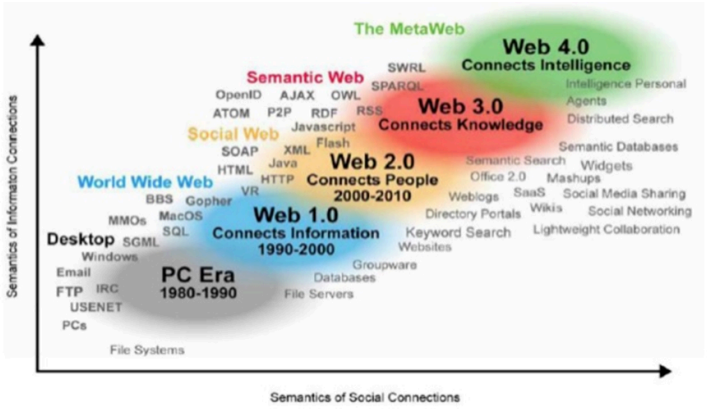
Una versión adicional del estándar HTML, la 3.0, apareció para dar respuesta a nuevas incorporaciones y propuestas de más etiquetas procedentes de distintas empresas, como Nestcape y o Microsoft.

Paralelamente, otros lenguajes para Internet fueron apareciendo, como **PHP (Hypertext Preprocessor)** y **ASP (Active Server Pages)**, que se propusieron para funcionar con bases de datos y aprovechar la interactividad que puede ofrecer la Web.

HTML 4.0 se publicó el 24 de Abril de 1998, entre sus novedades destacaron la utilización de hojas de estilos CSS, la posibilidad de incluir pequeños programas o scripts, la mejora de la accesibilidad de las páginas diseñadas, tablas complejas y agilidad en los formularios. Con este nuevo estándar llegó una plataforma para entretener a los navegantes de Internet con juegos y aplicaciones desarrollados en otros lenguajes, como Java.

En el año 2004, las empresas Apple, Mozilla y Opera se organizaron para formar la asociación **Web Hypertext Application Technology Working Group (WHATWG),** cuya actividad se centra en el actual estándar **HTML5**. HTML5 es una realidad actualmente y ya son muchos los navegadores importantes que lo implementan. HTML5 es una revolución en el desarrollo web al integrar en el propio lenguaje de marcas tanto el diseño como la inclusión de elementos multimedia, entre otras funcionalidades. Un ejemplo de uso de HTML5, como sustituto de Adobe Flash para hacer animaciones y juegos, es Cut the Rope, un juego de ingenio implementado en 2012 como promoción de Internet Explorer.

La Web, igual que los lenguajes que se utilizan para su desarrollo, ha pasado por diferentes etapas y hay otras fases en el horizonte. En la siguiente figura se muestra la **evolución de la Web** atendiendo a retos que ésta pretende acometer. Así se habla de Web 1.0 (Web informativa), Web 2.0 (Web colaborativa), Web 3.0 (Web semántica) y Web 4.0 (Web ubicua).



En este punto, hay que entender que entre las competencias de un diseñador web no está solo en saber usar lenguajes de programación para crear sitios web, sino que es igual o más importante que el diseñador sepa plasmar lo que cada usuario quiere comunicar con su sitio web, además de dar **soluciones factibles** a los cambios que se puedan producir durante el desarrollo del sitio o posteriormente.

Por lo tanto, para el desarrollo de un sitio web no hay que comenzar a crearlo con el lenguaje de marcas HTML o cualquier otro. Antes de programar el diseñador debe **planificar**: debe saber qué quiere ofrecer y qué objetivos persigue el sitio web (qué quiere comunicar) y, posteriormente, trabajar la elaboración de distintas soluciones de diseño antes de decantarse por una opción u otra. En las siguientes secciones se muestran opciones para planificar correctamente un sitio web, de manera que los cambios que se quieran hacer sobre el diseño o los contenidos sea posible hacerlos en un tiempo adecuado.

7. Maquetación web. Elementos de ordenación

Por **maquetación web** se entiende la distribución, en el espacio considerado y disponible, de los elementos que conforman una página web. En otras palabras, maquetar es colocar las diferentes partes de una página dentro de sus límites. Evidentemente, esta idea está muy ligada con lo visto en las secciones anteriores.

La ventaja principal de maquetar es **mantener separado** el contenido de la página de la presentación, es decir, que si hay cambios en los contenidos no tenga que tocarse el diseño y viceversa. De este modo, se hace más sencillo el mantenimiento y los cambios al contenido y diseño que se tengan que hacer. Sin embargo, ésta no es la única ventaja. Por el lenguaje que se usa actualmente para la maquetación de sitios web, ésta es interesante para reducir el tiempo de desarrollo y el tiempo que el usuario debe esperar a que se cargue completamente el sitio.

Hace unos años, la maquetación de las páginas web se realizaba **utilizando tablas** (etiquetas <table><tr><td> de HTML). Una vez entendido este proceso podía resultar sencillo, aunque si no se dominaban las tablas, podía convertirse en algo tedioso. El problema de las tablas es que generaban una página muy encorsetada y el código se volvía complejo de entender. Además, algunos buscadores encontraban problemas al analizar la estructura de la página y su uso puede causar problemas de accesibilidad e interpretación de los contenidos organizados en ellas.

Actualmente, la maquetación con tablas ha caído en desuso y se realiza **utilizando capas** (etiqueta de HTML), también llamadas **divisiones o contenedores**. La colocación de capas en una página web se realiza a través de **hojas de estilo o CSS**, concepto que es clave para entender el diseño actual de los sitios web. Las capas permiten, por ejemplo, que se pueda pasar de un diseño con un menú lateral a otro con el menú en la parte superior, solo cambiando la hoja de estilos.

7.1. Distribución de elementos en la interfaz: capas, marcos

En esta sección se describen dos de los elementos clave en la maquetación web: **capas** y **marcos**.

Las **capas**, también llamadas **DIV o layout**, son como contenedores donde se colocan imágenes, textos o incluso, otras capas. Las principales características de las capas son las siguientes:

· Las capas pueden estar **anidadas**, es decir, pueden estar unas dentro de otras. Básicamente, lo que se hace es definir cómo se posiciona en la página web, su colocación y su tamaño.

· Las capas son bloques con **contenido HTML** que pueden posicionarse de manera dinámica y anidarse. Las ventajas que ofrecen las capas solo se pueden aprovechar al cien por cien utilizando **estilos CSS**.

En realidad, las capas no se definen completamente mediante el lenguaje HTML, sino que necesitan del lenguaje de definición de estilos CSS. Con uno y otro lenguaje se pueden incluir en las páginas web elementos movibles, ocultables y, en general, manipulables de forma dinámica.

Por ejemplo:

```
<STYLE TYPE="text/css">
    .CapaFija {
        position: absolute;
        top: 200px;
        left: 20px;
        width: 200px;
        height: 100px;
        background-color: lightgray;
    }
</STYLE>
```

Con esto hemos definido un tipo de capa, denominada CapaFija, cuya altura es de 100 píxeles (un píxel depende de la resolución y tamaño de la pantalla, por ejemplo, en una pantalla de 1024 x 768, un píxel en horizontal son 0,329 mm y en vertical son 0,35 mm) y la anchura de 200 px. Además, está situada a 200 px de la parte superior y a 20 px del margen izquierdo de la página. Con el código anterior se ha **definido** una **clase .CapaFija**, pero todavía no se ha construido la capa. Para **construirla** se utiliza la etiqueta <DIV> y el **atributo class**, tal y como se muestra a continuación:

```
<DIV class="CapaFija">
    <H1>Esto es contenido</H1>
    <P>Aquí sigue más contenido HTML</p>
</DIV>
```

También se podría definir simplemente un **estilo** con **#CapaFija** y aplicarlo al <DIV> con el **atributo ID**.

```
<STYLE TYPE="text/css">
    #CapaFija {
        position: absolute;
        top: 200px;
        left:20px;
        width:200px;
        height: 100px;
        background-color: lightgray;
    }
</STYLE>
<DIV ID="CapaFija">
    <H1>Esto es contenido</H1>
    <P>Aquí sigue más contenido HTML</p>
</DIV>
```

Cualquier bloque <DIV> con ID="CapaFija" estará en esa posición y con ese tamaño.

Esta capa puede colocarse en cualquier parte de la ventana, su posición es **absoluta** (absolute). Pero también podemos definir capas de posicionamiento **relativo**, es decir, que más que definir las coordenadas de su posición respecto a la ventana, describimos su posición respecto al lugar donde aparezca en el texto. En otras palabras: describimos el desplazamiento de la capa respecto de donde la ponemos. Se definen así:

```
    <STYLE TYPE="text/css">
        .CapaFija {
            position: absolute;
            top: 200px;
            left: 20px;
            width: 200px;
            height: 100px;
            background-color: lightgray;
        }

        .CapaRelativa {
            position: relative;
            left: 150px;
            top: 100px;
            background-color: blue;
        }
    </STYLE>
    <DIV class="CapaFija">
        <H1>Esto es contenido</H1>
        <P>Aquí sigue más contenido HTML</p>
    </DIV>
    <P>Texto para relativizar</p>
<!--según exista o no este texto el siguiente div cambiará su posición
relativa -->
    <DIV class="CapaRelativa">
        <H1>Esto es contenido</H1>
        <P>Aquí sigue más contenido HTML</p>
    </DIV>
```

Además de las capas, otra alternativa para la maquetación son los **marcos (frames),** representados en HTML con etiqueta <frameset> y <frame>. Su uso mueve y ha movido controversia entre algunos diseñadores y adhesión por parte de otros. Los marcos son una forma de **insertar varias páginas web en una sola**. Los marcos dividen la página web en varias partes y dentro de cada parte se incluye otra página web. La idea es parecida a las capas, pero dentro de cada marco en vez de haber texto, imagen u otra capa, hay una página web.

Mal utilizados pueden arruinar la mejor página web, puesto que la pantalla del monitor está físicamente limitada. Cada marco que compone la página poseerá sus propios bordes y barras de desplazamiento, comportándose como **ventanas independientes**. Su situación en la página es rígida, no podemos colocarlos en las posiciones que deseemos, si tenemos cuatro marcos se situarán en cada uno de los cuatro cuadrantes de la pantalla. Si tenemos dos la pantalla se dividirá en dos filas o en dos columnas para alojarlos.

El problema principal de los marcos es que algunos navegadores no lo pueden manejar. Esto requiere que el diseñador, mediante código incrustado en la página, controle esta posible situación. El siguiente código utiliza **marcos fijos** e incluye código para controlar limitaciones del navegador:

```
<HTML>
    <HEAD>
        <TITLE>Los frames: páginas multiventana</TITLE>
    </HEAD>
    <FRAMESET COLS="20%,80%">
        <FRAME NAME="indice" SRC="indice.html">
            <FRAME NAME="principal" SRC="principal.html">
                <NOFRAMES>
                    <P align="center">Al parecer tu navegador no soporta
marcos, actualizate.</P>
                </NOFRAMES>
    </FRAMESET>
</HTML>
```

Dentro de un <FRAMESET> se definen los marcos que componen el conjunto y la acción alternativa para navegadores que no soporten marcos. A cada uno de los marcos se le adjudica un nombre y se especifica qué página HTML se mostrará en él (etiqueta <FRAME>). En el ejemplo solo queda definir lo que verá el usuario en el supuesto de que su navegador no soporte marcos (etiqueta <NOFRAMES>).

Comparado con las capas, los marcos requieren de mucha habilidad en el diseño para ser la base de una maquetación. Sin embargo, en algunas situaciones pueden ser muy útiles. Por ejemplo, si se desea compartir un cierto contenido por todo el sitio web, la barra de navegación o la cabecera, los marcos son de gran ayuda.

Una excelente alternativa a los marcos fijos son los **marcos flotantes**, actualmente soportados por todos los navegadores. La idea de este elemento, ideado por Microsoft, sigue siendo la misma: incluir una página externa dentro de otra, pero en este caso el marco puede quedar totalmente integrado en la página contenedora.

La programación con frames ya no se utiliza en el desarrollo web moderno, quedando obsoletos por varias razones:

- Problemas de usabilidad
- SEO limitado
- Compatibilidad limitada
- Mantenimiento difícil

En lugar de usar frames, hoy en día se emplean otros enfoques que son más robustos, compatibles y eficientes:

- **CSS** y **Flexbox/Grid**: Para dividir una página en varias secciones o "ventanas", hoy en día se usa CSS con técnicas como Flexbox o CSS Grid. Estas permiten crear layouts responsivos y mucho más flexibles sin los inconvenientes de los frames.
- **iframes**: Aunque no es ideal en muchos casos, los iframes (<iframe>) pueden usarse si necesitas incrustar contenido externo (como un video o un mapa) dentro de una página. Sin embargo, su uso debe ser limitado.
- **Aplicaciones de una sola página (SPA):** Con frameworks modernos como **React**, **Vue.js**, y **Angular**, es común crear aplicaciones de una sola página donde solo ciertas partes de la página se actualizan dinámicamente sin recargar toda la página. Esto es mucho más eficiente y flexible que usar marcos.
- **Plantillas dinámicas con JavaScript**: La mayoría de las páginas web modernas cargan contenido dinámico utilizando **JavaScript** y **AJAX**, permitiendo la actualización de una parte de la página sin recargar todo el documento.

Una página puede rechazar o **bloquear la conexión desde un iframe** como medida de seguridad. Esto se hace para evitar que otros sitios web incrusten su contenido sin permiso, lo que podría comprometer la seguridad o la privacidad del contenido.

8. Mapa de navegación. Prototipos.

Los sitios web pueden contener muchas páginas, todas ellas accesibles desde algún punto del sitio y todas con todos o algunos enlaces a las demás. Esta estructura de enlaces hace, en muchos casos, difícil que el diseñador o el usuario del sitio sepan qué páginas llevan a cuáles. Por ello, antes de diseñar un sitio web se debe realizar un **esquema** que permita anticipar cuáles son las secciones en las que estará dividida el sitio web y la relación entre los diferentes bloques de contenidos. Ese esquema recibe el nombre de **mapa de navegación** y es algo parecido al índice de contenidos de un libro, es decir, una manera de que el diseñador de un sitio web estructure bien los contenidos antes de crear el sitio y de que los usuarios encuentren más rápidamente lo que buscan una vez creado el sitio.

La siguiente figura muestra un ejemplo de mapa de navegación. El mapa muestra cómo están relacionados los diferentes grupos de información.

Existen seis tipos o clases de mapas de navegación

1. **Navegación lineal**: Es muy útil cuando se quiere llevar un proceso paso a paso. Podemos ir a la siguiente página o a la anterior.

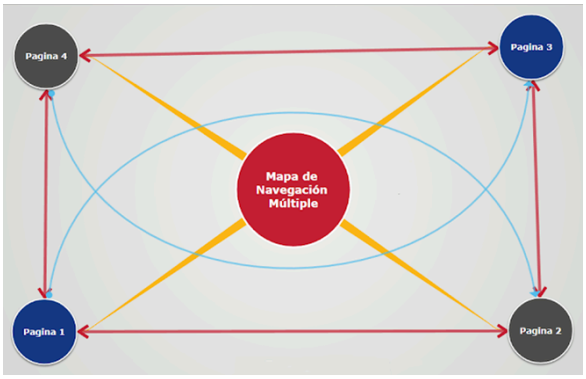
2. **Navegación lineal en estrella:** Es igual al anterior, pero este va y vuelve al inicio.

3. **Navegación Jerárquica:** Comienza con una página principal o raíz, se presentan varias opciones que permite ir visualizando páginas más específicas.

4. **Navegación No Lineal:** Tiene que conservar el camino general, pero hay que dar cabida a ligeras variaciones.

5. **Navegación Compuesta**: Combina distintos tipos de Mapas de Navegación.

6. **Navegación Múltiple**: Es La estructura de un sitio en el que cada una de sus páginas está vinculada a todas las demás.



A la hora de realizar mapas de navegación hay que tener en cuenta que las páginas que forman un sitio web no deben aportar todas ellas la misma información ni cubrir el mismo objetivo, se podrá partir de una **página principal (home o homepage)** y, desde ella, poder acceder al resto de páginas que conforman el sitio web.

Además de los mapas de navegación, la propia complejidad del diseño hace que, en muchos casos, sea difícil entender qué es lo que el usuario quiere transmitir al resto del equipo de desarrollo y qué es lo que tienen que hacer. Por tal motivo, los **prototipos** son herramientas muy interesantes para ahorrar tiempo a la hora de determinar qué es lo que hay que hacer, ya que muestran un esquema de cómo quedará el sitio web, pero empleando mucho menos tiempo que si hubiese que hacer hacerlo realmente.

Más detalladamente, un prototipo web es un borrador o modelo inicial a partir del cual se empieza a concebir y desarrollar la idea original del diseño de un sitio web. Hacer un prototipo es **más sencillo y económico** que hacer una web real y luego modificarlo hasta alcanzar lo que se busca. El prototipado de las páginas web resuelven básicamente los siguientes aspectos:

- **Qué elementos** deben conformar la interfaz de cada página.
- Qué elementos o características serán **comunes** a lo largo de las distintas páginas del sitio web.
- **Cuántos elementos** deben conformar la interfaz para que haya suficiencia en la información/interacción, pero evitando la saturación de elementos (de cada página).
- **Cómo debe organizarse** el mapa de navegación (en qué orden y disposición van las páginas).
- **Qué extensión** (superficie visual o tamaño) adecuada deben tener aprovechando eficientemente el espacio bidimensional disponible.
- **Qué aspectos** deben tenerse en cuenta a la hora de desarrollar el sitio web. Entre los aspectos especialmente interesantes, por su repercusión en los usuarios finales que usen o visiten un sitio web, están los aspectos técnicos, de usabilidad y de accesibilidad.

A la hora de realizar prototipos se puede separar la interfaz gráfica en dos **grupos de elementos o componentes:**

- Los **elementos o componentes abstractos** y comunes a toda página web, como pueden ser las cabeceras, barras de navegación (vertical u horizontal), los pies de página, los formularios, etc.
- Los **elementos concretos** específicos de una parte o del total de una página web, que se utilizan con un objetivo y una apariencia concreta, por ejemplo, botones, enlaces, campos de texto, imágenes, texto, etc.

Una vez que el diseñador ha realzado prototipos y el mapa de navegación, y estos han sido aprobados, el siguiente paso es abordar el desarrollo más detallado de un sitio web. En este paso se discuten otras características de los sitios web, como son las ligadas a la maquetación o el diseño detallado de dichos sitios. Para ello se puede trabajar con plantillas ya elaboradas o trabajar en el desarrollo de una propia. A este aspecto volveremos más adelante en este mismo capítulo.

9. Interpretación de guías de estilo. Elementos

Por su complejidad, para diseñar eficazmente interfaces web, son necesarias dos actividades: la **planificación** de qué se quiere hacer y la **coordinación** del equipo de desarrollo que se encarga del diseño. Una técnica para el diseño de sitios web dentro de entornos de desarrollo es la creación de una **guía de estilo**.

La guía de estilo es un documento (o varios) que define las pautas y normas de calidad que debe seguir una interfaz web para un determinado sitio web. Gracias a la guía de estilo se garantiza la coherencia del sitio, integrando toda la interfaz con un aspecto y uso homogéneos. La guía de estilo abarca aspectos de calidad de uso, accesibilidad, diseño gráfico, marketing, etc., tocando temas como los colores y otros elementos de diseño, como estándares (de usabilidad, accesibilidad, etc.). Más concretamente, se puede decir que una guía de estilo para la interfaz de usuario sirve como:

- Una herramienta para **garantizar la coherencia** de un sitio web a través de las páginas web del sitio.
- Una técnica para conseguir **integrar a todos los miembros** de un equipo de trabajo en un mismo objetivo, ya que se establecen las pautas que todos deben seguir. Además, ayuda a la formación de nuevos miembros de un equipo de trabajo.

La coherencia tiene varias interpretaciones: coherencia con las expectativas del usuario, coherencia en todos los sitios web que están relacionados, coherencia en todos los sitios web que no están relacionados pero que provienen de la misma empresa, coherencia con las normas de facto (por ejemplo, el uso de enlaces azules para denotar los enlaces no visitados), coherencia de la terminología, coherencia de la interacción, coherencia visual, coherencia entre las páginas/diálogos/ventanas, coherencia en el uso de los iconos o coherencia de los mensajes de error.

No existe una estructura única que deban seguir las guías de estilo. Sin embargo, algunas de las preguntas que debe responder son: ¿Qué colores tendrá la web y tonos? ¿Qué fuentes se usarán? ¿Qué formato de fuente se usará para los títulos, subtítulos, encabezados y el texto principal? ¿Cuál será la estructura? ¿Habrá encabezado, pie de página o menús? ¿Habrá un menú o varios? ¿Cuántos y dónde colocarlos? ¿Qué imágenes se mostrarán? ¿Dónde se colocarán? ¿Habrá logotipo? ¿Dónde se colocará? ¿Se tratarán la accesibilidad de la página y criterios de calidad de uso?

Dados los elementos que forman una guía de estilo se quiere hacer énfasis en su utilidad para todos los participantes en un desarrollo web, usuarios, desarrolladores o, incluso, el propio negocio vinculado al sitio web. Estos beneficios están relacionados con aspectos como: reducir cambios, facilitar el uso o mejorar la coordinación del equipo de trabajo.

10. Aplicaciones para desarrollo web

En la actualidad son muchas las **herramientas** que pueden utilizarse para facilitar tareas relacionadas con la planificación, diseño, desarrollo y mantenimiento de sitios web. El diseñador web no suele partir de cero, sino que se apoya en estas herramientas para desarrollar un sitio, ahorrando esfuerzo y focalizándolos en aspectos menos automatizables.

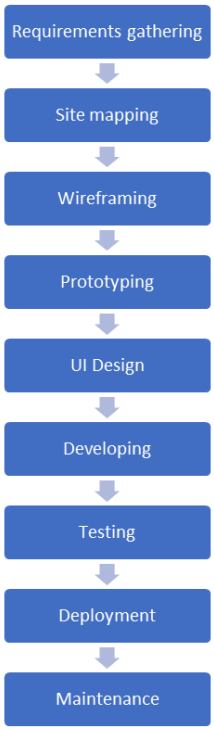
Las herramientas para desarrollar sitios web pueden clasificarse atendiendo a muchos **criterios**: su propósito, su coste, su alcance o la fase del proceso de desarrollo a la que den soporte, etc. Algunos grupos de aplicaciones útiles para abordar el desarrollo para la Web serían los siguientes:

- **General**: se incluirían en este grupo aquellos programas cuya utilidad es de interés general y de uso no solo exclusivo para los desarrolladores. Podrían incluirse en este apartado navegadores, herramientas software que faciliten realizar labores de planificación, tratamiento de imágenes o transferencia de ficheros (clientes FTP).
- **Diseño y prototipado**: serían aquellos programas útiles para diseñar páginas web. Diseño web y diseño en general. Dentro de estos programas, existen programas comerciales, no comerciales y programas que permiten hacer prototipos iniciales para discutir distintas posibilidades de diseño antes de llevar esos diseños a su programación final. Ejemplos de este tipo de programas podrían ser Balsamiq, MockFlow, Figma, Zeplin, Sketch, Adobe XD e InVision.
- **Multimedia**: serían programas orientados a la gestión o creación de animaciones y otros componentes con los cuales se puede dar más dinamismo a los sitios web desarrollados. Actualmente, las animaciones y gráficos en la web suelen implementarse con tecnologías nativas de HTML5, CSS3 y JavaScript, sin necesidad de plugins externos. Algunas herramientas son Adobe Animate, Greensock (GSAP), Three.js y Lottie.
- **Programación**: son programas enfocados a desarrolladores y programadores, con los cuales se elaboran páginas y sitios web. Los desarrolladores web utilizan una gran variedad de tecnologías para la creación de aplicaciones dinámicas y responsivas. Algunos lenguajes y frameworks utilizados son JavaScript (incluyendo frameworks como React.js, Vue.js, Angular), Node.js, TypeScript, PHP, ASP.NET Core, Ruby on Rails, y Python.
- **Editores y validadores HTML**: son programas para la edición de código HTML y para su comprobación, que ofrecen ayudas visuales específicas para construir webs, con validación en tiempo real y soporte para otras tecnologías. Algunas herramientas son Visual Studio Code, Sublime Text, Atom, Brackets y Pinegrow
- **Editores y validadores CSS**: programas que facilitan la creación, edición y comprobación de código CSS (hojas de estilo en cascada). Con nuevas características como Flexbox, Grid Layout, y CSS Variables. Algunas herramienta son Sass y LESS, Stylelint, Prepros, Figma y Sketch.

Dentro de estas herramientas también se encuentran los gestores de contenidos CMS que se explican en el siguiente apartado.

10.1. Fases en la creación de un producto digital

1. **Recopilación de Requisitos**: Se llevan a cabo reuniones con el cliente o los interesados para entender los objetivos del proyecto, las necesidades del usuario y los requisitos funcionales y técnicos. Esta fase es clave para definir la visión del producto.
2. **Arquitectura de la Información y Sitemap**: Se organiza la información y los contenidos que tendrá el producto, creando un **sitemap** para definir las secciones y su jerarquía. Aquí se establece cómo navegará el usuario a través del producto.
3. **Wireframing**: Se crean bocetos o **wireframes** que definen la estructura de las pantallas, posicionando los elementos principales sin detallar diseño visual (colores, imágenes, etc.). Se enfocan en la disposición y flujo de interacción.
4. **Diseño Visual (UI Design)**: A partir del wireframe, se trabaja en la estética del producto, definiendo paletas de colores, tipografías, iconografía y estilo visual general. Aquí se aplican principios de diseño para mejorar la experiencia del usuario y se obtiene un **mockup**, es decir, una representación visual detallada del producto final.
5. **Desarrollo (Frontend/Backend)**: Se transforma el diseño en un producto funcional. El frontend (lo que ve el usuario) se desarrolla junto con el backend (la lógica de negocio, bases de datos y servicios).
6. **Pruebas y Validación (Testing)**: Se realizan pruebas de usabilidad, funcionalidad y rendimiento para asegurar que el producto funciona según lo planeado y que cumple con los requisitos establecidos.
7. **Lanzamiento y Despliegue**: Se publica el producto, asegurando su correcto funcionamiento en el entorno final. Aquí también se define el plan de mantenimiento y mejoras continuas.
8. **Mantenimiento y Optimización**: Tras el lanzamiento, se monitorea el rendimiento del producto y se realizan actualizaciones y ajustes basados en feedback de los usuarios y análisis de datos.



10.2. Herramientas para wireframing y prototipado

Utilizada por profesionales de UX para diseñar y estructurar la interfaz de usuario de websites, aplicaciones y productos de software

- **Balsamiq** (https://balsamiq.com/):: Es una herramienta especializada en **wireframing**. Su enfoque es crear esquemas básicos y rápidos, sin muchos detalles visuales, para centrarse en la estructura de las interfaces. No se enfoca tanto en prototipos interactivos. Vídeos: Overview
- **MockFlow** (https://mockflow.com/): Es otra herramienta centrada en el **wireframing**. Permite diseñar estructuras de interfaz y tiene algunas capacidades básicas para la creación de prototipos interactivos. Overview
- **Figma** (https://www.figma.com/): Es una herramienta poderosa tanto para **wireframing** como para **prototipado**. Puedes crear wireframes simples y luego evolucionar a prototipos interactivos completos, incluyendo la colaboración en tiempo real y el diseño de interfaces con detalle visual. Overview
- **Zeplin** (https://zeplin.io/): Aunque no es una herramienta de wireframing ni de prototipado como las otras, Zeplin se utiliza para **handoff**, que es la fase donde los diseños (mockups o prototipos) creados en herramientas como Figma o Sketch se comparten con los desarrolladores. Facilita la colaboración entre diseñadores y desarrolladores proporcionando especificaciones y guías de estilo. Overview
- **Sketch**: Muy usada por diseñadores para crear interfaces web y móviles.
- **Adobe XD**: Herramienta para diseño y prototipado de interfaces.
- **InVision**: Para crear prototipos interactivos y colaborar con equipos de desarrollo.

10.3. Herramientas multimedia

- **Adobe Animate** (https://www.adobe.com/es/products/animate.html): Aunque Flash ha sido descontinuado, Animate ahora permite crear animaciones HTML5. Overview
- **Greensock - GSAP** (https://gsap.com/): Una de las bibliotecas de animación más potentes para JavaScript. Overview
- **Three.js** (https://threejs.org/) Biblioteca de JavaScript para gráficos 3D en la web, ideal para crear contenido interactivo.
- **Lottie** (https://lottiefiles.com/es/): Para animaciones ligeras en formato JSON que pueden renderizarse fácilmente en la web. Overview

10.4. Lenguajes y frameworks de programación

- **JavaScript** (incluyendo frameworks como **React.js**, **Vue.js**, **Angular**).
- **Node.js**: Para desarrollo del lado del servidor.
- **TypeScript**: Un superset de JavaScript que añade tipado estático.
- **PHP**, **ASP.NET Core**, **Ruby on Rails**, y **Python** (con frameworks como **Django** o **Flask**) siguen siendo relevantes.
- **Next.js**, **Nuxt.js** (para desarrollo full-stack con React y Vue).
- **GraphQL**: Para consultas más eficientes y flexibles en el backend.

10.5. Editores y validadores

Editores y validadores HTML:

- **Visual Studio Code**: Uno de los editores de código más populares, con extensiones para HTML, CSS, y JavaScript, entre otros lenguajes.
- **Sublime Text**: Un editor ligero y potente.
- **Atom**: Otro editor open-source que soporta HTML y otros lenguajes.
- **Brackets**: Un editor web-friendly con previsualización en vivo y desarrollado por Adobe.
- **Pinegrow**: Un editor visual de páginas web que soporta edición WYSIWYG (What You See Is What You Get, lo que ves es lo que consigues).

Editores y validadores CSS:

- **Sass y LESS**: Preprocesadores CSS que hacen que la escritura de CSS sea más potente y modular.
- **Stylelint**: Herramienta moderna para la validación de código CSS.
- **Prepros**: Una herramienta para compilar preprocesadores como Sass, Less y también manejar otros aspectos del desarrollo front-end.
- **Visual Studio Code**: Con extensiones para la validación y previsualización de CSS.
- **Figma y Sketch**: Aunque son herramientas de diseño, facilitan la creación y exportación de estilos CSS desde prototipos y mockups.

11. Generación de documentos y sitios web

Como se ha mencionado, crear sitios web no siempre requiere empezar desde cero. Con la llegada de los **Gestores de Contenidos**, conocidos como **CMS** (Content Management Systems), muchas empresas e instituciones optan por construir sitios web utilizando estas plataformas. En la actualidad, algunos de los CMS más utilizados a nivel global y en España son WordPress, que domina el mercado gracias a su facilidad de uso y amplia personalización, Joomla y Drupal, que siguen siendo opciones populares para proyectos más personalizados y robustos. Además, Headless CMS como Strapi y Contentful han ganado relevancia, especialmente en aplicaciones modernas que requieren mayor flexibilidad en la gestión del contenido.

Un gestor de contenidos se define como una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera **independiente el contenido y el diseño**. De esa manera, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores.

El éxito de los gestores de contenido radica principalmente en que **alejan los aspectos técnicos** de desarrollo del diseño de la interfaz y, ambos, de la generación de la información y documentación que se quiere comunicar en el sitio. De alguna manera, se puede afirmar que, con un gestor de contenidos, un administrador puede crear contenidos sin necesidad de saber nada más allá que manejar un procesador de textos.

Los gestores de contenidos más extendidos tradicionalmente están basados en **tecnologías web** como PHP/HTML y suelen utilizar gestores de bases de datos como MySQL o MariaDB. Ejemplos como WordPress, Joomla, y Drupal utilizan estas tecnologías, y muchos de ellos son de código abierto con licencias libres (por ejemplo, Joomla usa GPL). Sin embargo, en la actualidad también se utilizan otros lenguajes y tecnologías, como JavaScript (en CMS headless como Strapi), Node.js, o bases de datos NoSQL como MongoDB. Estos CMS permiten una gran flexibilidad y pueden ser adaptados para satisfacer las necesidades específicas de un negocio, ya sea mediante plugins o integraciones personalizadas.

Más concretamente, los gestores de contenidos guardan tanto los elementos de las páginas web como las especificaciones del diseño en bases de datos. La construcción de un sitio web se hace utilizando elementos de diseño predefinidos, llamados **plantillas**. Todos los elementos son leídos desde la base de datos, cargados automáticamente, puestos en el sitio preciso del diseño y presentados al usuario como página web. Esto garantiza aislar el diseño de los contenidos y la distribución de los componentes, pudiendo así cambiar el diseño sin tocar ninguno de los otros aspectos.

12. Plantilla de diseño

Las plantillas de diseño web son una opción popular para crear sitios web atractivos y profesionales **sin necesidad de una gran inversión de tiempo o recursos**. Estas plantillas, que ya vienen con una **estructura prediseñada**, permiten a los usuarios enfocarse en personalizar el contenido y las páginas, sin preocuparse por el aspecto visual y el código subyacente.

Hoy en día, las **plantillas** no solo están disponibles en plataformas tradicionales CMS como WordPress o Joomla, sino también en **constructores de sitios web** (website builders) como Wix, Squarespace, o Shopify, que ofrecen flexibilidad adicional.

Sin embargo, las plantillas son más adecuadas para sitios web con **estructuras simples o medianas**, como portafolios, blogs o sitios corporativos que no requieren funcionalidades complejas. Para proyectos que exigen personalización avanzada o integraciones específicas, los diseños a medida siguen siendo preferidos.

13. Interacción persona-ordenador

Una vez vistos los elementos principales que afectan al diseño de sitios web, en este punto se muestra el marco que afecta a todo desarrollo web que incluya interacción con los usuarios. La **Interacción Persona-Ordenador (IPO),** que así se llama ese marco, es la disciplina que estudia el intercambio de información entre las personas y los ordenadores (el término en inglés es **HCI, Human Computer Interaction**). Su objetivo es que este intercambio sea más eficiente, minimizando errores e incrementando la satisfacción. La IPO ofrece un marco empírico con todo lo referente a calidad de uso e interacción de interfaces de usuario. Aunque parezca que todo lo relacionado con interfaces de usuario sea algo exclusivo de técnicos en desarrollo de aplicaciones o ingenieros relacionados con la Ingeniería del Software, realmente, la IPO es la disciplina que sin duda más tiene que decir en este campo.

Los orígenes de la IPO hay que buscarlos en la rama de la **Psicología Aplicada** que estudia la Interacción Persona-Ordenador. La Psicología es la disciplina que estudia la percepción, la memoria, la adquisición de habilidades y el aprendizaje, la resolución de problemas, el movimiento, las tareas de juicio, de búsqueda o procesamiento de información y de la comunicación, es decir, procesos todos cuyo conocimiento se requiere para el adecuado diseño de mecanismos de interacción del usuario. Aunque la **Psicología Cognitiva** es una ciencia muy joven en lo que respecta a investigaciones de carácter básico y sistemático, existen actualmente suficientes hallazgos basados en resultados empíricos que permiten el desarrollo de la IPO y, por ende, de sitios web adaptados a los usuarios. Actualmente, los estudios sobre **inclusión** y **accesibilidad** en la IPO tienen un enfoque central, dado que se busca crear interfaces que se adapten a personas con diferentes capacidades y contextos de uso.

La IPO es una disciplina sólida, cuyos estudios han permitido dar una base teórica al diseño y a la evaluación de aplicaciones informáticas (en los que se incluyen, claro está, el desarrollo web). La importancia de esta disciplina para cualquier diseñador/desarrollado web se pone sobre relieve al leer artículos sobre el tema escritos hace décadas en los que se predecían elementos de interacción de los que se disponen actualmente. Una de las asociaciones más influyentes en este campo es la **ACM SIGCHI** (Association for Computing Machinery's Special Interest Group on Computer-Human Interaction) que desde 1982 reúne a los mejores especialistas en IPO.

Para un diseñador web, es muy importante la labor que se hace dentro del marco de la IPO, ya que, gracias a las investigaciones punteras en este campo, se mejora día a día la manera de interaccionar con los sistemas informáticos. La lectura de publicaciones o estándares desarrollados dentro de la disciplina IPO permite al diseñador web estar actualizado con los avances de la disciplina y conocer el futuro de su profesión.

Las tecnologías y tendencias actuales, como la **experiencia de usuario** (**UX**), que ha cobrado más importancia en los últimos años, también está relacionada con la IPO. Otras herramientas y **técnicas modernas** que cabe mencionar son el prototipado, testing de usabilidad, o la **inteligencia artificial (IA)** aplicada a interfaces conversacionales y accesibilidad. El campo de la HCI se ha expandido a nuevas áreas, como la **realidad virtual (VR)**, **realidad aumentada (AR)** y el diseño de interfaces para dispositivos móviles y wearables, donde las interacciones con los usuarios son más complejas.