



Diseño de interfaces web

Unidad 1: Planificación de interfaces gráficas



ÍNDICE

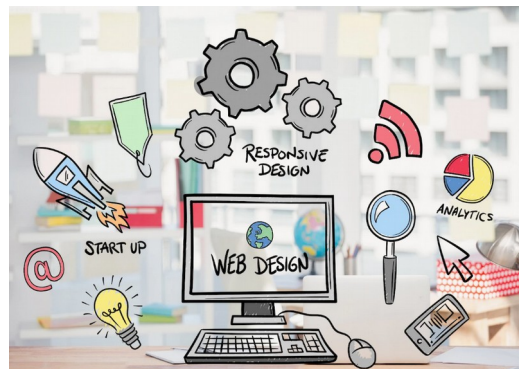
INTRODUCCIÓN.....	3
OBJETIVOS / CAPACIDADES.....	4
PROYECTO DE LA UNIDAD.....	5
1. ELEMENTOS DEL DISEÑO: PERCEPCIÓN VISUAL.....	8
2. TEXTURA (PUNTOS, LÍNEA, FORMA, VOLUMEN Y PROFUNDIDAD).....	10
Cuestionario.....	12
3. COLOR, TIPOGRAFÍA, ICONOS, LOGOS.....	13
4. INTERACCIÓN PERSONA-ORDENADOR, LA INTERFAZ DEL USUARIO. ESTILOS Y PARADIGMAS DE LA INTERACCIÓN PERSONA-ORDENADOR.....	15
5. INTERPRETACIÓN DE GUÍAS DE ESTILO. ELEMENTOS.....	17
Cuestionario.....	19
6. DETECCIÓN DE PATRONES. PATRONES DE DISEÑO WEB.....	20
7. CARACTERÍSTICAS: USABLE, VISUAL, EDUCATIVA Y ACTUALIZADA.....	23
8. GENERACIÓN DE DOCUMENTOS Y SITIOS WEB. DOCUMENTACIÓN.....	25
9. CONCEPTO Y COMPONENTES DE UNA INTERFAZ WEB. CLASES, MÉTODOS, PROPIEDADES Y EVENTOS EN LA INTERFAZ WEB.....	27
Cuestionario.....	29
10. ZONAS Y SISTEMA DE NAVEGACIÓN, CONTENIDO E INTERACCIÓN.....	30
11. APLICACIONES PARA DESARROLLO WEB. CONTROL DE VERSIONES.....	32
12. LENGUAJES DE MARCAS.....	35
13. MAPA DE NAVEGACIÓN.....	37
14. MAQUETACIÓN WEB. ELEMENTOS DE ORDENACIÓN.....	39
Cuestionario.....	41
15. TABLAS.....	42
16. CAPAS. ARQUITECTURA DE LA NAVEGACIÓN.....	44
17. MARCOS. DISTRIBUCIÓN DE ELEMENTOS EN EL INTERFAZ: MARCOS EN LÍNEA.....	46
18. PLANTILLA DE DISEÑO.....	48
Cuestionario.....	50
RESUMEN.....	51
RECURSOS PARA AMPLIAR.....	52
BIBLIOGRAFÍA.....	53

GLOSARIO.....	54
---------------	----

INTRODUCCIÓN

Este módulo trata de una de las cosas a las que quizás se le da poca importancia como programadores, pero realmente de cara al usuario es de las más relevantes.

Muchas veces pensamos que un buen desarrollador es aquel que crea un código eficiente, rápido, sencillo y comprensible. Y en parte es así, pero cuando hablamos de **programación visual** o **programación web** (en este caso), el **apartado gráfico** es fundamental, ya que es con lo que interacciona el usuario.



Durante esta unidad veremos una pequeña introducción al mundo de las **interfaces gráficas**, viendo los elementos del **diseño**, de qué se compone una **interfaz**, o algunos **patrones de diseño**.

En esta unidad, y en las siguientes se recomienda tener **conocimientos** tanto de **programación** como de **lenguajes de marcas**, especialmente **HTML**, ya que va íntimamente relacionado con el lenguaje CSS que se verá a lo largo del módulo.

OBJETIVOS / CAPACIDADES

En esta unidad de aprendizaje, las capacidades que más se van a trabajar son:

- ✓ Identificar y aplicar parámetros de calidad en los trabajos y actividades realizados en el proceso de aprendizaje para valorar la cultura de la evaluación y de la calidad y ser capaces de supervisar y mejorar procedimientos de gestión de calidad.



PROYECTO DE LA UNIDAD

Antes de empezar a trabajar el contenido, te presentamos la **actividad** que está relacionada con esta unidad de aprendizaje. Se trata de un **caso práctico** basado en una **situación real** con la que te puedes encontrar en tu puesto de trabajo. Con esta actividad se evaluará la puesta en práctica de los **criterios de evaluación** vinculados al resultado de aprendizaje que se trabaja en esta unidad. Para realizarla deberás hacer lo siguiente: lee el enunciado que te presentamos a continuación, dirígete al área general del módulo profesional, concretamente a la actividad de evaluación que se encuentra dentro de esta unidad, allí encontrarás todos los detalles sobre fecha y forma de entrega, objetivos... A lo largo de la unidad irás adquiriendo los conocimientos necesarios para ir elaborando este proyecto.

Enunciado:

¿Qué vemos hoy?

Nos ha gustado tanto el proceso de creación del boceto y primer diseño de nuestra página web, que finalmente nos hemos decidido a **crear nuestra primera web** que sirva como portfolio para una posible entrevista de trabajo.

Hemos pensado en la opción de usar una **plantilla predefinida**, y modificarla, para hacerlo todo un poco más sencillo.

La idea es hacer la cartelera de una empresa de cines, usando la plantilla:

https://www.w3schools.com/w3css/tryw3css_templates_pizza.htm.

Tenemos la suerte de que hemos escogido una plantilla existente, open source y encima responsive. Por tanto, lo único que tendremos que hacer es adaptarla a nuestro propósito. Haremos la cartelera de 3 cines, uno en Madrid, uno en Barcelona y uno en Sevilla.

Se trata de:

- Mantener la **barra de navegación**, pero cambiando:

- HOME por INICIO.
 - MENU por SALAS.
 - ABOUT por SERVICIOS.
 - CONTACT por TAQUILLA.
- Cambiar el título por "**CINES DIW**", y el subtítulo por "**Tus cines preferidos**".
 - Cambiar la **imagen de la cabecera** por una más adecuada.
 - La antigua estructura del **MENU** (ahora llamada **SALAS**) debe tener en **cada pestaña uno de los 3 cines**. El usuario podrá "navegar" entre las 3 pestañas y se le mostrarán las películas de cada cine.
 - **Crea** por lo menos **5 películas diferentes** en cada cine, y en cada película:
 - Donde antiguamente se mostraba el precio de la pizza, ahora debe mostrarse el nombre de la sala (Sala 1, Sala 2, etc.).
 - Donde estaba el nombre de la pizza, ahora debe estar el título de la película.
 - Donde estaban los ingredientes de la pizza, ahora se debe mostrar la hora u horas de proyección de las películas.
 - El antiguo **apartado ABOUT** (ahora llamado **SERVICIOS**) debe explicar qué **servicios** tiene el cine: aparcamiento, aseos, venta de golosinas, etc. Cambia las imágenes por las que creas más adecuadas.
 - El antiguo **apartado CONTACT** (ahora llamado **TAQUILLA**) debe mostrar un **formulario de venta de entradas**, donde:
 - Habrá un selector de cine.
 - Habrá un selector de sala.
 - Habrá un selector de hora.
 - Habrá un `<input>` de tipo numérico para preguntar el número de entradas.
 - Habrá un botón llamado "Comprar".

- Cambia el **color del menú principal y del botón**, de negro a verde.
- Todo lo que no se ha especificado a los requisitos lo podrás montar como quieras, pero intenta **diseñar un prototipo adecuado** a la temática a desarrollar.
- Recuerda que estás desarrollando un **prototipo** que de momento no es necesario que sea funcional, pero sí que debe tener una **estética correcta** para poder ser presentado como **portfolio**.
- **Documentate y valora qué IDE utilizarás** para realizar los cambios. Debes argumentar la decisión.

1. ELEMENTOS DEL DISEÑO: PERCEPCIÓN VISUAL

En el mundo actual, tendemos a dar mucha importancia al **diseño** de las cosas, ropa, muebles, relojes, electrodomésticos... En ciertos aspectos esto suele estar relacionado más bien con la estética, pero cuando hablamos de diseño de **interfaces gráficas**, va mucho más allá.



DESTACADO

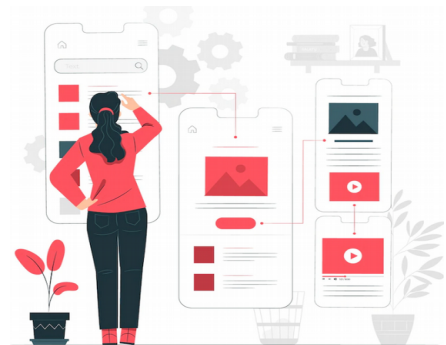
Podemos definir una **interfaz gráfica**, como la capa que permite la **relación** entre el **usuario** y la **máquina**. Cuando hablamos de programación web, la interfaz sería la propia web, y, por lo tanto, su diseño no sólo afecta de manera estética, si no funcional.

El hecho de que un sitio web esté bien diseñado afecta en cómo el usuario puede **interactuar** con él.

- ➔ La **percepción visual** es la sensación que tiene un usuario cuando está viendo una **página web**. Lo que se pretende con el **diseño** de la página es transmitir al usuario esa percepción.

No podemos diseñar de igual manera una página de un periódico que una página destinada al público infantil. Tanto el diseño como los propios **elementos** de la **interfaz** han de ser diferentes, porque van destinados a un público diferente.

De ahí que el desarrollador haya de tener todos estos factores en cuenta a la hora de **diseñar un sitio web**.



- ➔ En multitud de ocasiones se ve al desarrollador como una persona con poco sentido de la **estética** o el **diseño**, e incluso en el campo de la programación hay gente que se dedica en exclusiva al **código** de la aplicación o al diseño de ésta.



**PRESTA
ATENCIÓN**

*No obstante, es importante conocer **ambos campos**, para poder decantarse por uno, o simplemente poder combinarlos en nuestros proyectos. Todo esto es lo que intentaremos conseguir a lo largo del módulo.*

2. TEXTURA (PUNTOS, LÍNEA, FORMA, VOLUMEN Y PROFUNDIDAD)

Uno de los aspectos más importantes del **diseño** es aquel que ni siquiera es visible **de cara al usuario**, o al menos directamente.



*Hablamos de los **elementos conceptuales**, que como decía el pintor Wucius Wong, no son visibles, y de hecho no existen, pero parecen estar siempre presentes.*

*El hecho de que los elementos estén alineados, que guarden una cierta equidistancia, que tengan un aspecto similar... Pueden ser invisibles a ojos del usuario de a pie, pero son vitales en la tarea del **diseñador**.*

Podemos definir, pues, **cinco elementos conceptuales básicos**:

- Punto.
- Línea.
- Forma.
- Volumen.
- Profundidad.



A continuación, pasamos a definir estos **cinco elementos básicos**:

➔ Punto.

Un punto indica una **posición**. No es algo en sí visible, aunque tradicionalmente lo imaginemos como un pequeño círculo. No tiene medida ni ocupa un espacio.

➔ Línea.

Es otro elemento que no es visible por sí mismo, sino que es la **unión de dos puntos**. Conceptualmente, una línea tiene una longitud, pero no una anchura. Sirve para delimitar espacios o formar los bordes de un plano.

➔ Forma.

Una forma es el resultado de la unión de **varias líneas**. Mediante ellas se pueden crear formas regulares o irregulares, polígonos... Pero todo quedaría aún en dos dimensiones.

→ **Volumen.**

Entramos ya en las **tres dimensiones**. Para formar elementos con volumen necesitaríamos juntar varias formas, en diferentes planos, de tal manera que conseguimos una forma tridimensional.

→ **Profundidad.**

Indica el **plano** en el cual se posiciona una forma o volumen. A mayor profundidad, más alejado parece de la vista del observador, y a la inversa.

Cuestionario



Lee el enunciado e indica la opción correcta:

¿Cuál de los siguientes elementos conceptuales del diseño no tiene medida ni ocupa un espacio?

- a. Línea.
- b. Forma.
- c. Punto.



Lee el enunciado e indica la opción correcta:

¿Cuál de los siguientes elementos del diseño se define como la unión de dos puntos y tiene longitud pero no anchura?

- a) Forma.
- b) Volumen.
- c) Línea.



Lee el enunciado e indica la opción correcta:

¿Cuál de los siguientes elementos del diseño implica la creación de formas tridimensionales al combinar varias formas en diferentes planos?

- a) Punto.
- b) Profundidad.
- c) Volumen.

3. COLOR, TIPOGRAFÍA, ICONOS, LOGOS

Vistos los elementos conceptuales, que permiten de alguna manera estructurar el diseño de una interfaz, aunque no sean propiamente visibles, nos centraremos ahora en los **elementos visuales**.

A menudo, como usuarios, entendemos estos elementos como los más importantes a la hora de crear un diseño, ya que son los que realmente podemos ver.

Aquí podríamos hablar de:

- Colores.
- Tipografías.
- Iconos.
- Logos.



Veamos en profundidad estos **conceptos**:

→ Colores.

Todo lo que percibimos en nuestra **vida** tiene un cierto color, con algunos **matices, contrastes** o **tonalidades**. A menudo podemos asociar los colores con objetos, productos o incluso marcas comerciales.

→ Tipografías.

Nos referimos a la tipografía como el **tipo de letra** que podemos utilizar, desde la casi omnipresente Arial a la muy desenfadada (quizás demasiado) Comic Sans. La tipografía influye en gran medida en el **aspecto** de una página web.

→ Iconos.

Un icono es una **representación gráfica** de algún elemento de la vida real, una manera de representar una **acción** o una **idea**. Un claro ejemplo es el icono de guardar, que se representa mediante un viejo disco flexible.

→ Logos.

Normalmente sirven para representar **productos** o **marcas** comerciales. Su diseño ya puede dar para mucha literatura, por lo que no será objeto de estudio de este módulo.

A continuación, te proponemos la visualización del siguiente **vídeo** sobre la **configuración de paletas de colores**:

Vídeo: configuración de paletas de colores



Visualiza este vídeo en el que hablamos sobre la configuración de paletas de colores.

4. INTERACCIÓN PERSONA-ORDENADOR, LA INTERFAZ DEL USUARIO. ESTILOS Y PARADIGMAS DE LA INTERACCIÓN PERSONA-ORDENADOR



Hemos definido las **interfaces gráficas** como la herramienta que permite la **interacción** entre el **usuario** y la **máquina**, o, dicho de otra manera, entre la persona y el ordenador.

Estas **interfaces** permiten al usuario **comunicarse con la máquina** para que ésta realice las acciones adecuadas y retorne resultados de nuevo al usuario. Así que la interfaz permite generalmente una comunicación de **ida y vuelta**.

Durante todo el módulo nos basaremos en las **interfaces web**, y por tanto en la comunicación con el usuario mediante el diseño de las páginas web.

No obstante, existen otros **tipos de interacción persona-ordenador**.

Los **paradigmas actuales** más importantes son:

- El **ordenador personal**.
- La **realidad virtual**.
- La **realidad aumentada**.

Profundicemos en el conocimiento de estos **paradigmas**:



➔ El ordenador personal.

Es el tipo de interacción con una máquina más común. Se trata de hacer que el usuario controle un **ordenador** para realizar las tareas deseadas.

Pese a que lo definimos como uno de los paradigmas, actualmente lo podríamos ampliar con la interacción con **otros dispositivos**, como teléfonos móviles, tabletas gráficas, consolas o incluso algunos electrodomésticos.

➔ La realidad virtual.

La realidad virtual o **VR** es un término que se usa para situar al usuario en una **realidad alternativa**, diferente a la real, en la que se puede interactuar con ciertos elementos, pero estos nunca son reales, aunque se asemejen bastante.

La interacción en la realidad virtual se realiza de una manera parecida a la que lo haríamos en el mundo real.

→ **La realidad aumentada.**

Muchas veces se confunde realidad virtual con realidad aumentada o **AR**, pero en realidad son términos muy diferentes.

La realidad aumentada trata precisamente de eso, de **complementar el mundo real**, con imágenes o sensaciones virtuales, de manera que se mejora la interacción.

Actualmente tenemos muchos usos de realidad aumentada, para el control de tráfico, propósitos médicos, entretenimiento...

5. INTERPRETACIÓN DE GUÍAS DE ESTILO. ELEMENTOS



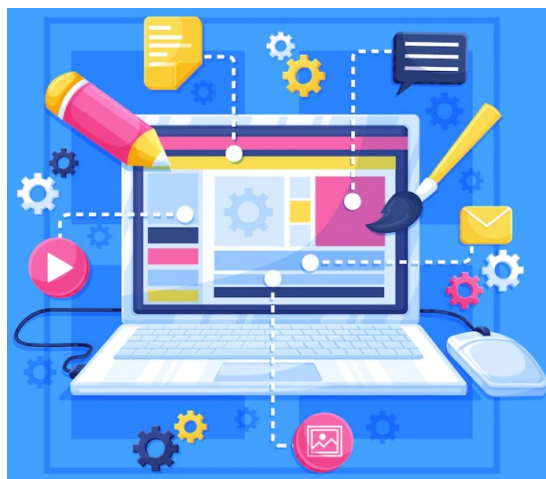
DESTACADO

*Podemos definir una guía de estilos como un conjunto de **reglas** o **recomendaciones** tipificadas a la hora de crear una interfaz de usuario, o en nuestro caso una interfaz web.*

En estas guías se pueden **especificar cosas como:**

- Paleta cromática.
- Tipografías.
- Botones y controles.
- Imágenes.

Veamos de manera más concreta **en qué consisten estos elementos:**



→ **Paleta cromática.**

Dicho de otra manera, los **colores** que se van a usar en el sitio web. Es importante que estos colores tengan una cierta correspondencia entre ellos, para hacer el diseño más **agradable y legible**.

Pese a eso puede haber diferencias en las tonalidades o incluso en los propios colores, dependiendo de su uso, si se trata de colores de letra o fondo, etc.

En la actualidad muchas interfaces tienen tanto el **modo claro** como el **oscuro**, por lo cual es necesario tener esto en cuenta a la hora de diseñar la paleta cromática del sitio web.

→ **Tipografías.**

Formas, tamaños y colores de los tipos de letras. También puede haber diferencias dependiendo de su uso, pero al igual que con los colores se recomienda cierta concordancia entre ellas.

El uso del **tipo de letra** marca de gran manera el aspecto de nuestra web, por lo que nos permite hacerla más formal, divertida o elegante con tan solo cambiarlo.

→ **Botones y controles.**

Define el **estilo de los controles** que forman nuestro sitio web, como las **cajas de texto**, los **botones**, los **menús desplegables**.

Aquí podríamos también incluir los **iconos** usados para indicar ciertas acciones o atajos, que también tendrían que seguir un mismo estilo.

→ **Imágenes.**

Básicamente el **tamaño y las proporciones** de las imágenes que se usarán en el sitio web. Pero también se podría incluir aquí si las imágenes tienen algún tipo de borde, transparencia, color de fondo...

Además, es posible que ciertas imágenes requieran ciertas variaciones, por tratarse de imágenes destacadas, cabeceras, etc.

Cuestionario



Lee el enunciado e indica la opción correcta:

En la realidad aumentada...

- a. Se interactúa con un ordenador de sobremesa.
- b. Se complementa al mundo real con imágenes virtuales.
- c. Se muestra una realidad alternativa.



Lee el enunciado e indica la opción correcta:

¿Cuál de los siguientes no es un elemento conceptual de diseño?

- a. Punto.
- b. Profundidad.
- c. Color.



Lee el enunciado e indica la opción correcta:

¿Cómo se denomina al conjunto de colores de una página web?

- a. Paleta cromática.
- b. Tipografía.
- c. Sistema de navegación.

6. DETECCIÓN DE PATRONES. PATRONES DE DISEÑO WEB



PRESTA
ATENCIÓN

*Crear una página web es como empezar un cuadro, con un lienzo en blanco. Antes de empezar una composición, tenemos que tener en cuenta el **espacio** del que podremos disponer y qué **elementos gráficos y textuales** se utilizarán.*

*Una vez sabido esto, deberemos definir cuáles son las **partes de la pantalla** que estarán **ocupadas** y qué **elementos posicionaremos**. Para eso hay muchos **factores** a tener en cuenta.*

Si dejamos muchos **espacios vacíos**, la composición parecerá **descompensada**, ya que será mucho más difícil relacionar los elementos entre ellos.

Por el contrario, si el **número de elementos posicionados** es elevado, podemos obtener un **diseño muy sobrecargado** y difícil de entender, en el que no se verá nada clara la finalidad de la página y qué pasos hay que hacer para lograrla.

Además, en todo momento tenemos que tener en cuenta la agrupación lógica de los elementos. Es decir, tener al alcance aquellos que están relacionados entre sí.



DESTACADO

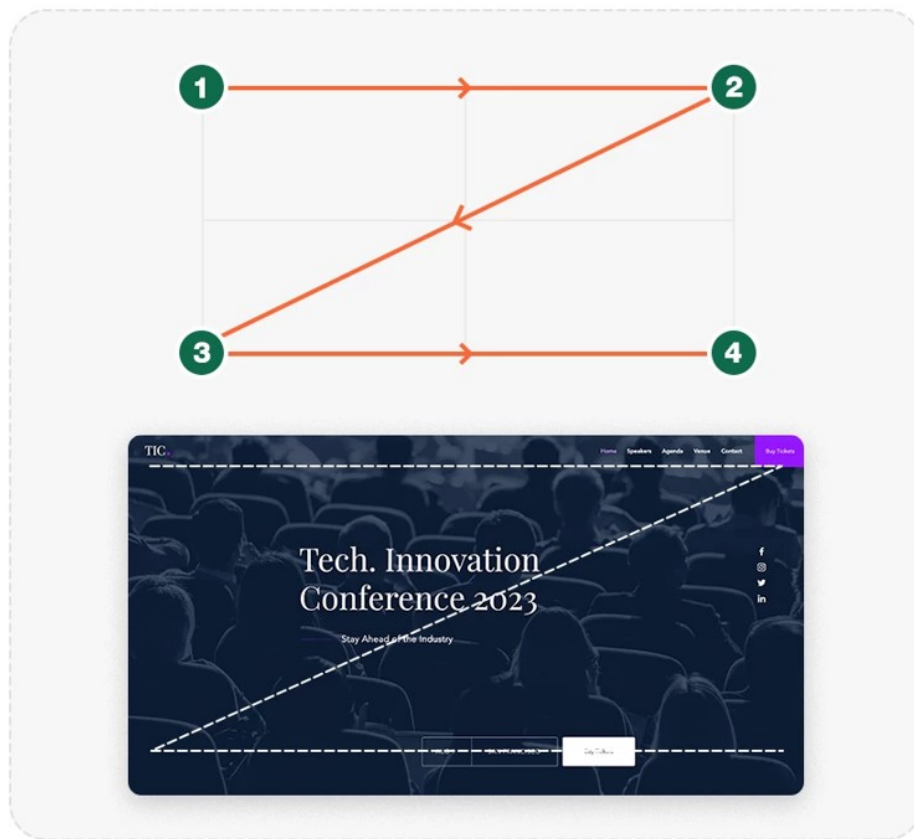
*La **composición** se define como una **distribución equilibrada** de todos los elementos incluidos en un diseño. En este sentido, la composición del diseño de un sitio web debe ser **estética** y **funcional**.*

Con el fin de obtener una composición adecuada, existen lo que se denomina **patrones de diseño**. Podemos encontrar múltiples patrones, pero dos de los más utilizados los veremos a continuación.

→ Diseño con patrón Z.

Básicamente es la **disposición de los elementos en el orden habitual de lectura**. Es decir, empezamos por arriba y vamos leyendo a hacia la derecha, hasta que bajamos a la siguiente línea y repetimos el proceso, **en forma de Z**.

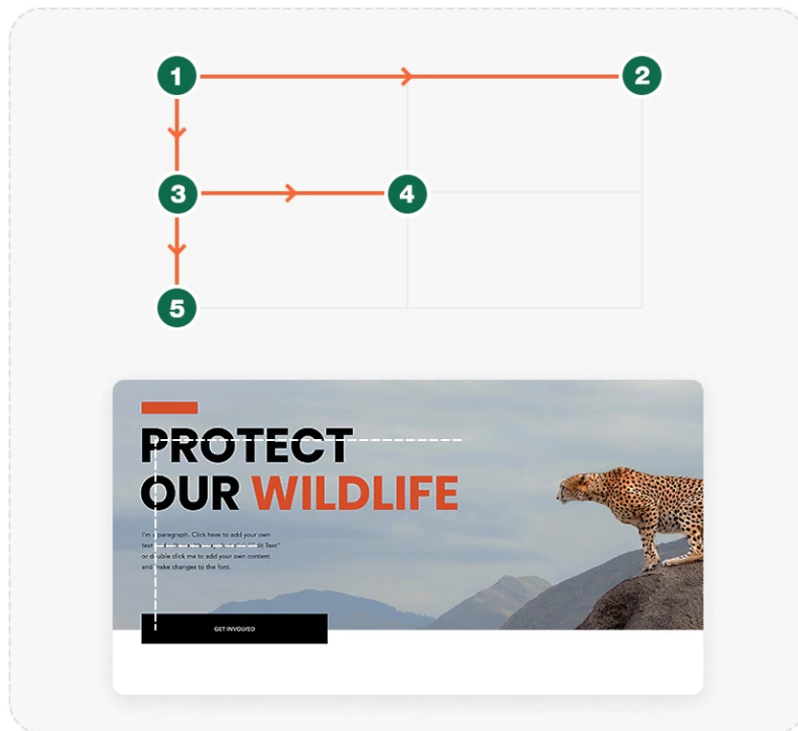




Este sería útil para **sitios web donde hay muy poco contenido escrito**, por ejemplo, para una página principal.

➔ **Diseño con patrón F.**

En este caso **los elementos se colocan en forma de F**, con claro predominio de elementos en la columna izquierda, que se van desarrollando hacia la derecha.



Esto permitiría al usuario **escanear de forma rápida páginas con mucho contenido escrito**, sin tener que leerlo todo necesariamente.

Vídeo: patrones de diseño web



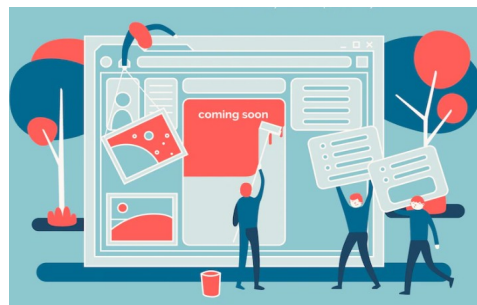
Visualiza el siguiente vídeo sobre los patrones de diseño web:
<https://www.youtube.com/embed/25RvbkQ1QM>

7. CARACTERÍSTICAS: USABLE, VISUAL, EDUCATIVA Y ACTUALIZADA

Para que una página web esté bien construida, es decir, tenga un **diseño** en el que el **usuario** se pueda sentir **cómodo** para pasar un buen rato, se deben de cumplir una **serie de características**.

Es aquí donde hablamos que una **web debería ser**:

- Usable.
- Visual.
- Educativa.
- Actualizada.



Profundicemos en estas **características básicas** que debe tener toda página web:

→ Usable.

Se dice que una web es **usable** cuando permite que el usuario pueda interactuar con ella de manera **sencilla** y **efectiva**. Básicamente se trataría de poner las cosas fáciles al usuario, y que no tenga que perder mucho tiempo en la página para poder realizar tareas sencillas.



En la parte final de este módulo dedicaremos una unidad íntegra a la usabilidad, por lo que no nos extenderemos mucho más ahora.

→ Visual.

Una web visual favorece que el usuario se sienta cómodo, mediante los **recursos gráficos**. No sólo las imágenes o iconos, si no la distribución de los elementos incluidos en la página.

→ Educativa.

Se dice que una web es educativa cuando su **curva de aprendizaje** tiene poca pendiente. Es decir, al usuario le resulta fácil aprender a manejar la página, y

es capaz de retener estos conocimientos, para cuando vuelva a acceder después de un tiempo.

➔ **Actualizada.**

Por último, es importante que una página web permanezca **actualizada** por parte de los creadores, para no tener una sensación de abandono, que haga que el usuario deje de navegar por ella.

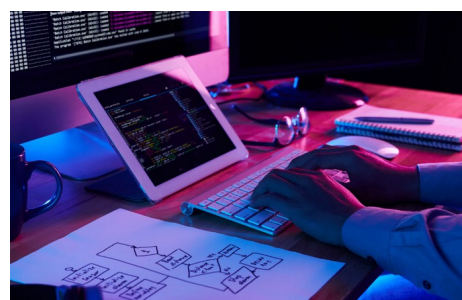
Es importante que no sólo el contenido esté actualizado, si no que el propio diseño de la web siga las tendencias del momento.

8. GENERACIÓN DE DOCUMENTOS Y SITIOS WEB. DOCUMENTACIÓN



Actualmente el **desarrollo web** está tan extendido que prácticamente cualquiera puede tener un conocido capaz de crear una página web. Pero la realidad es que la generación de sitios web es un **proceso más complicado** de lo que pueda parecer.

Mucha gente parece dedicarse al desarrollo web, en lo que es una de las profesiones con más intrusismo, pero no son muchos los que dominan todo el **proceso de creación**. Nosotros como futuros buenos desarrolladores, hemos de conocerlo.



La **generación de páginas web**, como casi todas las ramas de la programación, **consta de varias fases**, que juntas conforman todo el proceso.

Así podemos hablar de:

- ➔ **Análisis**: es la **fase inicial** del desarrollo. En ella se reúnen el **cliente y el profesional** para **valorar las ideas del primero**, ver cuáles se pueden llevar a cabo y cuáles no, y de qué manera.
- ➔ **Diseño**: una vez el desarrollador tiene claro lo que quiere el cliente, llega el momento de **diseñar** el sitio web. Para ello se crean **bocetos, mockups y prototipos**. Se escogen los colores, las tipografías, los estilos. Pero no sólo eso, que sería en cuanto al diseño, si no que se ponen las **bases** de lo que será el **código de la web**.
- ➔ **Desarrollo**: es la **parte principal** del desarrollador, ya que las dos anteriores se pueden llevar a cabo por analistas o diseñadores. El **desarrollo** del sitio web implica en su esencia escribir **código**, ya sea **HTML** o **CSS** para la estructura y diseño, o **JavaScript** o **PHP** para los scripts.

- **Pruebas y depuración:** una vez creado nuestro sitio web hay que **probarlo**. Aquí puede intervenir el propio desarrollador, o incluso probadores (externos e internos), que **testearán** las versiones **alfa o beta**. Una vez probado, y si se detectan **errores**, se podría **volver a fases anteriores**, como el análisis o el desarrollo. Es lo que se conoce como desarrollo en cascada.
- **Documentación:** como buenos desarrolladores deberíamos conocer la importancia de una **buena documentación**. Puede ser **interna** (del código) o **externa** (de utilización de cara al cliente). Lo ideal es realizar esta fase en paralelo con el desarrollo.
- **Puesta en producción:** finalmente, cuando está listo, es el momento de **lanzar el sitio web**, o ponerlo en producción, para que sea utilizado por los usuarios finales.

Actividad de aprendizaje 1: creación de página web

Estás tan entusiasmado con el nuevo módulo de diseño de interfaces web que se te ha ocurrido una idea genial: **crear una pequeña web** para usarla como portafolio de cara a una posible entrevista de trabajo.

Por desgracia aún no dominas demasiado el tema, por lo que crees más oportuno **trabajar con un compañero**.

La idea es **crear una página web** de una empresa imaginaria (que quizás sea la tuya propia en un futuro).

Como sabes, lo primero que necesitáis para ello es **crear un pequeño boceto o mockup**.

Esta primera actividad trata de buscar la **mejor herramienta de mockups** para diseño web, que creáis que mejor se adapta a vuestras necesidades, realizando una **comparativa** entre las distintas opciones.

Podéis buscar información por separado, para posteriormente **debatirla** en común y **buscar la mejor alternativa**. Entrega el **archivo .pdf** en el espacio habilitado para ello.

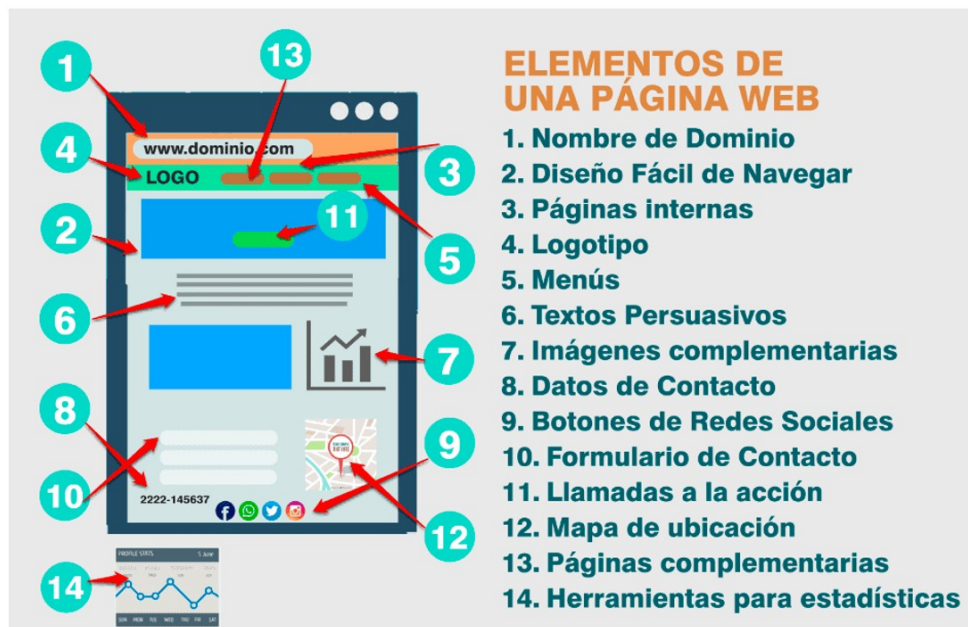
9. CONCEPTO Y COMPONENTES DE UNA INTERFAZ WEB. CLASES, MÉTODOS, PROPIEDADES Y EVENTOS EN LA INTERFAZ WEB



RECUERDA

Como hemos dicho, una **interfaz web** se compone de una **serie de herramientas y elementos** que permiten la **interacción** entre el ordenador, o en este caso un **servidor**, y el usuario.

Estos elementos pueden ser muy diversos, desde **imágenes** o **botones**, pasando por **menús**, **enlaces** o **barras de desplazamiento**. Sin embargo, todos ellos tienen que estar debidamente estructurados y agrupados en sus correspondientes zonas.



Así nos podemos encontrar los **elementos** de:

- ➔ **Identificación**: permiten al usuario saber **de quién o de qué es el sitio web**. Si es una empresa, una organización, un particular, y de quién se trata.
- ➔ **Navegación**: permiten al usuario **navegar** o saltar entre páginas, para encontrar lo que busca.

- **Contenidos**: muestra la **información** de la página web. El contenido puede ser de muchos tipos diferentes, texto, imágenes, multimedia.
- **Interacción**: permiten al usuario realizar ciertas **acciones** en la página.



De ellos hablaremos más en profundidad en el siguiente capítulo, pero es esencial tener algunas pinceladas previas.

Cuestionario



Lee el enunciado e indica la opción correcta:

Cuando podemos interactuar fácilmente con una web, se dice que es...

- a. Visual.
- b. Usable.
- c. Educativa.



Lee el enunciado e indica la opción correcta:

En el patrón de diseño F...

- a. Los elementos están ordenados de abajo a arriba.
- b. Los elementos siguen el orden natural de lectura.
- c. Se da importancia a la columna de la izquierda.



Lee el enunciado e indica la opción correcta:

En la fase de pruebas...

- a. Se lanzan versiones alfa y beta.
- b. Se crean documentación interna y externa.
- c. Se lanza la versión final.

10. ZONAS Y SISTEMA DE NAVEGACIÓN, CONTENIDO E INTERACCIÓN

Cuando estamos diseñando una página web nos encontramos con varias **zonas o apartados** importantes, que estarán, de una u otra manera en cada proyecto. Se trata de las **zonas de navegación, contenido e interacción**.

A continuación, profundizaremos para conocer con más detalle cada una de estas **zonas**:



→ Zona de navegación.

La **zona de navegación** es una de las herramientas imprescindibles en el diseño de cualquier sitio web. Por definición, un sitio web es un **conjunto de páginas web enlazadas entre ellas**, que permite la navegación de unas a otras.

Justamente por eso necesitamos un **sistema que permita esa navegación**. Que el usuario puede y sepa moverse entre los diferentes apartados de una web es parte esencial.



DESTACADO

*Los **elementos de navegación** tienen que ser lo suficientemente intuitivos para que el usuario sepa qué es lo que tiene que hacer en cada momento.*

*También es importante que haya algún **método para volver a la página inicial** o a la **parte de arriba de la página**.*

→ Zona de contenido.

Son las zonas en las que se muestra propiamente la **información** de la página.

Estas deben estar correctamente **estructuradas**, para dar al usuario una sensación de orden y coherencia. Lo podemos conseguir mediante **títulos, esquemas** o diferentes **tipografías**.



*No todas las páginas de nuestro sitio tienen por qué tener **zonas de contenido**, ya que algunas son exclusivamente para navegar por el sitio y no para mostrar información.*

→ Zona de interacción.

Por último, los **elementos de interacción** son aquellos que **permiten al usuario realizar acciones** en nuestra página web.

Aquí hablaríamos de los **botones, cajas de texto, menús desplegables...** Mediante ellos, el usuario podrá **interactuar** con la página, con el fin de obtener los resultados esperados.

11. APLICACIONES PARA DESARROLLO WEB. CONTROL DE VERSIONES

El **desarrollo web** se basa en un conjunto de **procesos** que permiten llegar al **diseño final** del sitio web. Estos procesos requieren de una serie de **herramientas o aplicaciones** para llevarse a cabo. Hablaremos de algunas de ellas a continuación.

→ Herramientas para la codificación.

Son aquellas herramientas que nos permiten **crear el código** de nuestra página web, ya sea el **HTML** o **CSS** que permiten su diseño como el código de script que hace la página dinámica.

Pese a que podemos conseguir esto simplemente con un editor de texto como el bloc de notas, hay **aplicaciones o IDEs** que hacen más fácil el desarrollo. Así podemos hablar, por ejemplo, del más utilizado a nivel de diseño como es **Dreamweaver**, de Adobe.

Para el código de script podríamos utilizar algún IDE, como por ejemplo **VSCode**, que además serviría también para programar la parte gráfica.

Con el paso de los años se han puesto de moda los **gestores de contenidos o CMS**, que a través de diseños y plantillas predefinidos permiten al desarrollador crear sitios web personalizados de manera más o menos sencilla. Los más utilizados serían **WordPress** o **Joomla**.

→ Herramientas para el diseño.

Aquí tenemos algunas aplicaciones que escapan a los contenidos de este módulo, pero que también conviene conocer, e incluso saber utilizar, como desarrolladores web. Hablamos de **editores gráficos** tan conocidos como **Photoshop** e **Illustrator**, de Adobe, o **Affinity Photo**, por ejemplo.

→ Navegadores.



Ya hemos visto que una parte importante del desarrollo es el **testeo de los sitios web**. Para eso, evidentemente necesitamos **navegadores web** en los cuáles poder ejecutarlos.

Cuantas más pruebas hagamos y en más navegadores diferentes, más probabilidades de éxito tendremos. Así que podemos utilizar los más conocidos como **Firefox, Edge, Chrome, Safari u Opera**.

→ **Control de versiones.**

Aquí sin duda alguna existe un rey desde hace unos años y no es otro que **Git**, en cualquiera de sus variantes.

Quizás la más conocida es la versión web, **GitHub**, que no sólo permite el control de versiones, ramas, y demás, sino que además permite la realización de código colaborativo, pudiendo compartir con otros usuarios o desarrolladores.

Te animamos a visualizar el siguiente **vídeo** para ampliar la información:

Vídeo: herramientas de desarrollo y diseño web



Visualiza este vídeo sobre herramientas de desarrollo y diseño web:

<https://www.youtube.com/embed/UfpkuE33YGQ>

Actividad de aprendizaje 2: creación del boceto de la página web

Tras realizar la comparativa, ya tenéis clara la mejor alternativa para la **realización del mockup** de vuestra primera página web. El siguiente paso es **crear el boceto** de lo que será la página principal.

Usad la **herramienta escogida para elaborarlo**, teniendo en cuenta colores, tipografías y demás.

Sobre todo, recordad que sólo es un **primer boceto**, no tiene que ser operativa.

Es importante que, a la hora de tomar las **decisiones de diseño**, éstas **sean puestas en común** entre los miembros del grupo, **llegando a un acuerdo** si hay posibles discrepancias.

Entrega el **archivo .pdf** en el espacio habilitado para ello.

12. LENGUAJES DE MARCAS

El **diseño y la estructura** de las páginas web está desarrollado en unos **lenguajes**, que no se pueden considerar lenguajes de programación como tal, ya que carecen de las estructuras que tienen éstos, como las variables, condicionales, bucles...



DESTACADO

*Hablamos de **lenguajes de marcas** como aquellos que permiten **estructurar una página o unos datos**, mediante **etiquetas**, normalmente de **apertura y cierre**.*

*Estos lenguajes poseen unas **etiquetas predefinidas**, cada una con una función, que permiten al desarrollador crear páginas web combinándolas.*

Básicamente existen **tres lenguajes de marcas** principales:

- HTML.
- XML.
- CSS.



Amplieemos la información de estos **tres lenguajes de marcas** para conocerlos más a fondo:

➔ HTML.

Acrónimo de **HyperText Markup Language** o **lenguaje de marcado de hipertexto**. Es el **principal lenguaje de marcas** para el **diseño de páginas web**.

Emplea **etiquetas y atributos** para definir la estructura de una página web, pudiendo definir apartados, links, formularios...

Normalmente se combinaría con algún **lenguaje de script**, como JavaScript, para darle un carácter dinámico a la página.

➔ XML.

Acrónimo de **eXtensible Markup Language** o **lenguaje de marcas extensible**. Es un lenguaje mucho más estricto que **HTML** en cuanto a su **estructura**.

El uso principal de **XML** es el **almacenamiento de datos**, de manera jerárquica, lo cual se consigue a través de sus etiquetas.

→ **CSS.**

Acrónimo de **Cascading Style Sheets** u **hojas de estilo en cascada**. Es un lenguaje que permite la **creación de estilos** para una página web.

Se **combina** también **con HTML** para definir el **diseño** de la página web, de manera sencilla y escalable. Lo conoceremos en profundidad en las siguientes unidades del módulo.

13. MAPA DE NAVEGACIÓN

Una de las tareas principales en el momento de diseñar un sitio web es la **elección de los contenidos** y, por eso, es necesario crear lo que se dice un **mapa de navegación** o **mapa del sitio**.



DESTACADO

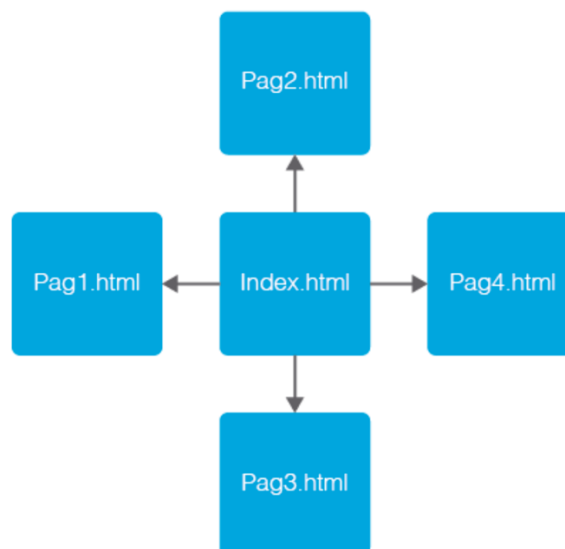
Se trata de una **lista de páginas web** estructuradas con una arquitectura de **enlaces** adecuada para una correcta **navegación** del usuario y un mejor **posicionamiento** en los motores de búsqueda.

Existen varios **tipos de mapas de navegación**:

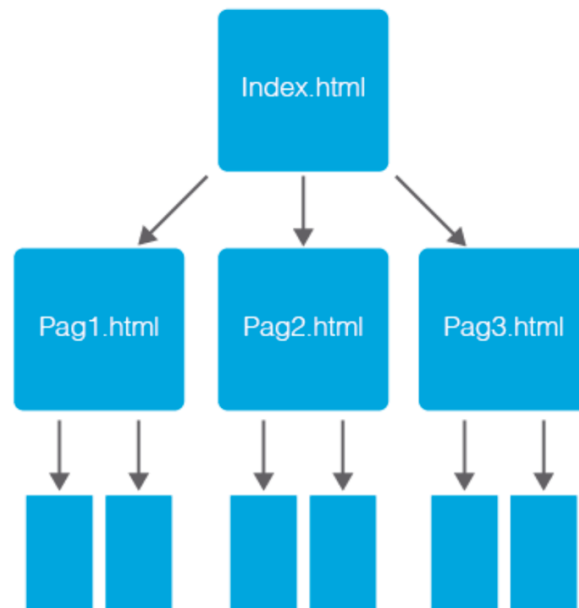
→ **Navegación lineal**: permite que el usuario reciba la información en el orden adecuado, únicamente con las opciones de **ir adelante o atrás**.



→ **Navegación lineal en estrella**: sistema de navegación similar al lineal, pero **se va y se vuelve al inicio**.



- ➔ **Navegación jerárquica o de árbol:** este sistema comienza por una **página principal** en la que se presentan varias opciones que permiten ir visualizando **páginas específicas**.



- ➔ **Navegación no lineal:** permite marcar un camino **general** de navegación, pero da cabida a pequeñas variaciones.
- ➔ **Navegación compuesta:** la navegación compuesta es la que incluye **distintos sistemas de navegación**.
- ➔ **Navegación múltiple:** esta estructura de navegación permite que cada página esté **vinculada** a todas las demás.

14. MAQUETACIÓN WEB. ELEMENTOS DE ORDENACIÓN

Como vimos anteriormente, una de las etapas más tempranas en el proceso de desarrollo de software en general y en el de web en particular, que es el caso que nos ocupa, es el **proceso de análisis**.



DESTACADO

*En esta etapa se tienen en cuenta las **ideas del cliente**, las cuales obtendremos mediante conversaciones. Éste nos presentará lo que **quiere** de la página web, una idea de **cómo tiene que ser**, lo que **tiene que hacer** y una serie de **funcionalidades**.*

Con todo esto, la primera **labor del desarrollador** será crear un **primer boceto de lo que sería la web**. Este proceso es el que se conoce como **maquetación**. Es decir, crear una maqueta, de igual manera que un arquitecto crea la maqueta de un edificio antes de ponerse a construirlo.



PRESTA
ATENCIÓN

*Maqueta, boceto o mockup son maneras de decir lo mismo. Y el primer paso que daríamos sería la **estructuración de los bloques** que tendrá el sitio web.*

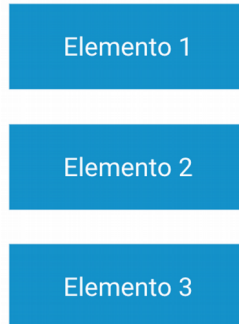
También tenemos que tener en cuenta el **diseño**, la **elección de colores**, **tipografías** y demás. De todas maneras, este boceto **no es algo definitivo**, por lo que se puede ir **redefiniendo o modificando** a lo largo del proceso de desarrollo.



Para realizar este boceto, y en general la página definitiva podemos utilizar algunos **elementos de ordenación**, que permiten la **distribución** de los diferentes elementos de la página.

Hablamos de **tres tipos principales de ordenación**:

→ **Elementos en bloque:** son elementos que ocupan todo el **ancho del contenedor** o padre.



→ **Elementos en línea:** se van posicionando en la **misma línea**, de izquierda a derecha.



→ **Elementos flotantes:** el elemento sale del **flujo normal** y puede aparecer a la izquierda o derecha de su contenedor.

Cuestionario



Lee el enunciado e indica la opción correcta:

¿Cuál es la zona dónde se permite realizar acciones al usuario?

- a. Navegación.
- b. Interacción.
- c. Contenido.



Lee el enunciado e indica la opción correcta:

¿Cuál es la herramienta por excelencia para el control de versiones?

- a. Illustrator.
- b. VSCode.
- c. Git.



Lee el enunciado e indica la opción correcta:

¿Cuál de los siguientes no es un lenguaje de marcas?

- a. PHP.
- b. XML.
- c. HTML.

15. TABLAS

Durante los primeros años de existencia de las páginas web, se utilizaba principalmente un método para distribuir la información de una página web, las **tablas**.

El uso de tablas permitía al desarrollador **posicionar los elementos** en diferentes posiciones, como si de una tabla de un procesador de textos se tratase.



A base de **filas y columnas** de diferentes tamaños se puede dar al contenido una estructura u otra.

También se puede jugar con los **bordes** y el **fondo de la tabla** para que esta estructura sea invisible a los ojos del cliente allí donde queremos que pase desapercibida.

Para ello, se utilizan una **serie de etiquetas HTML**, que permiten **definir la estructura**:

- ➔ **<table>** define todo el **contenido** de una tabla.
- ➔ **<th>** de Table Header, sirve como **encabezado** de una serie de celdas de la tabla.
- ➔ **<tr>** sirve para representar cada una de las **filas** que componen la tabla.
- ➔ **<td>** **representa el objeto final**, o sea cada una de las celdas de la tabla.



Veamos a continuación un **fragmento de código** en el que empleamos estas **etiquetas**:

Código: uso de etiquetas HTML

```
<table id="tabla1" border="1" style="width:100%;">
  <tr>
    <td>Concepto</td>
    <td>Precio</td>
    <td>Descripción</td>
  </tr>
  <tr>
    <td>Objeto 1</td>
    <td>AC10</td>
    <td>Descripción del objeto 1</td>
  </tr>
  <tr>
    <td>Objeto 2</td>
    <td>AC20</td>
    <td>Descripción del objeto 2</td>
  </tr>
</table>
```

A parte de esto, cada una de las etiquetas pueden tener **atributos**, que definen aún mejor sus **características**.



Al igual que con las tablas de los procesadores de texto, también podemos **combinar filas** o **columnas**. Para ello se usan los atributos "**colspan**" o "**rowspan**", que permite definir el número de filas o columnas que se combinarán.

16. CAPAS. ARQUITECTURA DE LA NAVEGACIÓN

A medida que ha ido pasando el tiempo, la estructuración de la página web en tablas ha ido dando paso a otro tipo de estructuras que permiten mejorar y facilitar el diseño.

Así, por ejemplo nos encontramos con las **capas**, que se definen mediante la etiqueta HTML `<div>`.



DESTACADO

Una **capa** no es más que un **elemento rectangular** dentro de la página en el cual podemos incluir cualquier tipo de contenido, ya sea texto, enlaces, imágenes.

Además, podemos **incluir tablas** o incluso **capas dentro de otras capas**, lo que se conoce como **capas anidadas**, para formar estructuras más complejas.

A continuación, veremos algunos **aspectos relevantes** sobre las **capas**:

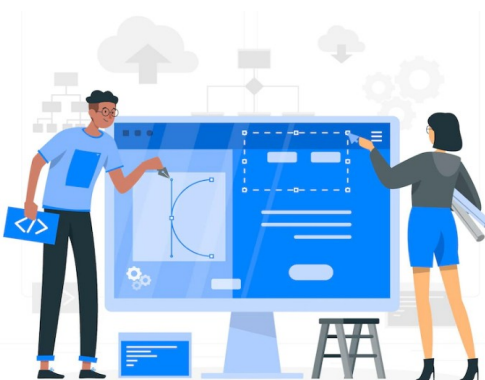
- ➔ La **principal ventaja de las capas** es que las podemos complementar con los estilos CSS (que veremos en unidades posteriores), para otorgarles un estilo, color de fondo, tipografía, márgenes...
- ➔ La **estructura** más simple, con las etiquetas de inicio y cierre de las capas sería:

```
<div> Bienvenidos </div>
```

- ➔ Además, podemos incluir **atributos** que permiten modificar el comportamiento de estas. Por ejemplo, darle un identificador mediante el atributo "id".

```
<div id="bienvenida"> Bienvenidos </div>
```

En este caso el **identificador** nos puede servir para **referenciar la capa** desde cualquier lenguaje de script, como por ejemplo JavaScript, y poder **sustituir o**



complementar su contenido con algo que hayamos calculado o creado a partir de alguna función.

Vídeo: trabajar con capas o DIV



Visualiza este vídeo en el que hablaremos sobre cómo trabajar con capas o DIV.

17. MARCOS. DISTRIBUCIÓN DE ELEMENTOS EN EL INTERFAZ: MARCOS EN LÍNEA

Otra de las herramientas que se usan a día de hoy para poder estructurar la información en una página web son los **marcos**.

De alguna manera, éstos sustituyen a las tablas, en temas de estructuración. De manera que las **tablas** a día de hoy se usan principalmente para **mostrar contenido relacionado**.

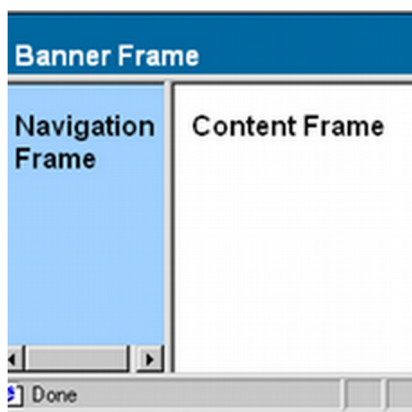


DESTACADO

Un **marco o frame** en HTML no es más que una **estructura** que nos permite **dividir la ventana de** nuestro navegador en **diferentes áreas**, cada uno con un tamaño específico, que puede ser fijo, o dinámico en función del tamaño del navegador.

→ Barras de desplazamiento.

Cada **marco es independiente** del resto, pudiendo incluso contener sus propias barras de desplazamiento si así lo queremos.



→ Etiqueta frameset.

Para la creación de una página web con marcos, debemos **sustituir la etiqueta HTML body por la etiqueta frameset**, pudiendo especificar, por ejemplo:

- ✓ Si se trata de **divisiones verticales u horizontales**, o incluso una mezcla de ellos.
- ✓ Las **medidas** de los distintos marcos de la página, que pueden ser fijas, o dinámicas según el porcentaje del espacio disponible, por ejemplo.

Cada marco tiene un **contenido diferente**, que vendrá indicado por la **etiqueta frame**, especificando la página HTML que lo contiene.

Ejemplo: código para la división de una página de manera horizontal con dos marcos

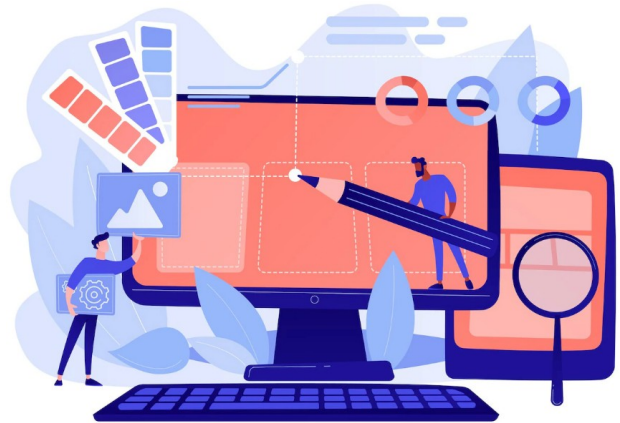
A continuación, veremos un **ejemplo de código** para la **división de una página** de manera horizontal empleando dos marcos:

```
<html>
<head>
  <title>Marcos</title>
</head>
<frameset rows="200,*">
  <frame src="pagina1.html" />
  <frame src="pagina2.html" />
</frameset>
</html>
```

En el ejemplo, dividimos una página de manera **horizontal**, con **dos marcos que hacen referencia a dos páginas HTML externas**. El marco de arriba tendrá 200 puntos de altura, y el de abajo ocupará todo el espacio que quede disponible en función de la altura total de la ventana (indicado con el asterisco).

18. PLANTILLA DE DISEÑO

Hemos visto durante los capítulos anteriores todo lo relativo a la **maquetación de las páginas** y cómo podemos estructurar los elementos mediante **tablas, capas o marcos**. Normalmente éste es un proceso que se hace de forma **manual** por el desarrollador o diseñador en función de las **demandas del cliente**, comentadas en la fase de análisis.



No obstante, para simplificar el diseño de las páginas, muchas veces se emplean lo que se denominan **plantillas de diseño**.

A continuación, veremos algunos **aspectos relevantes** sobre las **plantillas**:

→ **Funcionalidad.**



DESTACADO

Estas **plantillas** permiten crear un **diseño predefinido** que podemos personalizar a nuestro gusto. El **objetivo** de estas es **simplificar** mucho el proceso de creación de la página, agilizando todas las fases que son más o menos mecánicas, para poner énfasis en todo lo relativo al diseño y a la personalización.

→ **Dónde encontrarlas.**

¿Pero de dónde se obtienen esas plantillas? Existen muchos **sitios web** donde podemos conseguirlas, ya sean de forma gratuita o de pago. Incluso un desarrollador puede crear sus propias plantillas para reutilizarlas y adaptarlas en el futuro.



Sitios como **wix.com** o **webnode.es** permiten a un usuario que no es realmente desarrollador utilizar plantillas muy fácilmente adaptables para realizar su propia web.

Evidentemente, con esto conseguimos trabajos **poco profesionales** y **muy parecidos entre sí**. Por eso, son páginas que **no** utilizaría un desarrollador web profesional.

No obstante, si hablamos de **CMS**, como **WordPress** o **Joomla**, también encontramos el uso de plantillas, esta vez sí, que a menudo son **usadas por diseñadores y desarrolladores web**, debido a su alto potencial y posibilidades de personalización.



Además, un desarrollador experto puede conseguir resultados muy diferentes y que se adapten a las necesidades del cliente con una sola plantilla, por lo que se ahorra tiempo mediante la **reutilización de código**.

Actividad de aprendizaje 3: creación de la estructura de la web

Después de enseñar el boceto de la página web al profesor, éste os ha felicitado por el gran trabajo, y ahora que ya has avanzado un poco más en la unidad, te propone un **reto individual**.

Este reto consistirá en **pasar el boceto creado a una página web real**, creando la **estructura** mediante **tablas**, **capas** o lo que consideres necesario.

Seguirá sin ser una web completamente operativa, pero ya se podrá **ejecutar en un navegador**.

Entrega el **archivo .html** en el espacio habilitado para ello.

Cuestionario



Lee el enunciado e indica la opción correcta:

La etiqueta `div` sirve para definir...

- a. Una tabla.
- b. Una capa.
- c. Un marco.



Lee el enunciado e indica la opción correcta:

La etiqueta HTML `<tr>` sirve para...

- a. Definir la cabecera de una tabla.
- b. Definir una celda de una tabla.
- c. Definir una fila de una tabla.



Lee el enunciado e indica la opción correcta:

¿Con qué atributo definimos el identificador de una capa?

- a. `id.`
- b. `div.`
- c. `class.`

RESUMEN

Durante esta unidad, hemos estado viendo todo lo relativo al **diseño** y la **maquetación de páginas web**. Evidentemente, ha servido de introducción a lo que veremos a lo largo del curso, por lo que en muchos de los conceptos no hemos entrado en profundidad.

Hemos empezado viendo todo lo relativo al **diseño visual** en general, no necesariamente relacionado con el **diseño web**, para luego ver como lo usaremos al estructurar nuestras páginas.

Hemos incidido en la importancia de la **relación entre el ordenador y el usuario**, explicando con detalles las características de las **interfaces web** y las **guías de estilo**.

Además, hemos visto con detalle algunas **herramientas** que permiten la **estructuración** de los distintos **elementos que forman una página web**, desde los **patrones de diseño** a las **tablas** o los **marcos**, pasando por las más utilizadas a día de hoy, **capas**.

Finalmente hemos comentado el **uso de plantillas de diseño** y cómo pueden ayudar a los desarrolladores profesionales y no tan profesionales en el proceso de creación de una página web.

RECURSOS PARA AMPLIAR



PÁGINAS WEB

- 15 plataformas y programas para crear páginas web que debes conocer:
<https://mx.godaddy.com/blog/programas-para-hacer-paginas-web/> [Consulta septiembre 2022].
- Códigos de colores HTML: <https://htmlcolorcodes.com/es/> [Consulta septiembre 2022].
- Tipografías: qué son, los diferentes tipos y sus variables tipográficas:
<https://imborrable.com/blog/tipografias-que-son/> [Consulta septiembre 2022].



BIBLIOGRAFÍA



PÁGINAS WEB

- CSS. MDN: <https://developer.mozilla.org/es/docs/Web/CSS> [Consulta septiembre 2022].
- HTML: Lenguaje de etiquetas de hipertexto. MDN: <https://developer.mozilla.org/es/docs/Web/HTML> [Consulta septiembre 2022].
- W3C: <https://www.w3.org/> [Consulta septiembre 2022].



GLOSARIO

- **IDE:** entorno de desarrollo Integrado. Conjunto de herramientas que aúnan todo lo necesario para el desarrollador.
- **Mockup:** maqueta, boceto o prototipo de una aplicación.
- **PHP:** lenguaje de script de la parte servidor.
- **Script:** código que permite hacer dinámica una página web estática.
- **Versión alfa/beta:** versiones de prueba de una aplicación, abiertas a todo el mundo o a un número reducido de usuarios.

