



INSTITUTO de ENSEÑANZAS a DISTANCIA de ANDALUCÍA

## 2º de Bachillerato Tecnologías de la Información y Comunicación Contenidos

### Diseño web funcional.Prevencción y buenas prácticas: Introducción a la programación en entorno cliente.Estándares

En temas anteriores del curso hemos estudiado la realización de páginas web de contenido estático con lenguaje **HTML** en estado puro y cómo podíamos mejorar su estética y apariencia gracias las hojas de estilo (**CSS**). Pero **HTML&CSS** se pueden complementar con otros lenguajes para poder realizar acciones más complejas permitiendo que su contenido se genere o modifique de forma autónoma o como resultado de la interacción con el usuario. Dicho de otro modo: **podemos dotarlo de comportamiento a partir de scripts**.

**¿Qué aprenderemos en este tema?** Conoceremos que significa "**programación en entorno cliente**" y uno de los lenguajes más importantes hoy en día, **JavaScript**. ¿Recuerdas las plantillas web realizadas a partir de framework como "**Bootstrap**"? Las plantillas web hacen uso de Javascript para otorgar funcionalidad a la página. Facebook, Twitter, y muchas otras aplicaciones hacen uso de JavaScript. Descomprime [ejemplo](#) y ejecuta en tu navegador index.html te harás una idea.



Imagen en pixabay de [FirmBee](#) bajo licencia [Creative Common](#)



Imagen en flick de [Maximiliano](#) bajo licencia [Creative Common](#)

Además, analizaremos dos conceptos estandarizados fundamentales en el desarrollo web:

- **"La usabilidad"**: los usuarios deben ser capaces de alcanzar sus objetivos con un mínimo esfuerzo y unos resultados máximos, aprenderemos las pautas para conseguirlo.
- Sin olvidar la idea de hacer la web accesible para todos los usuarios independientemente de las circunstancias y los dispositivos utilizados a la hora de acceder a la información, **"la accesibilidad"**.

## Desarrollo Web

La web fue inicialmente concebida y creada por Tim Berners-Lee. Dicho prototipo estaba basado en el concepto de hipertexto creándose unos protocolos y especificaciones que han sido adoptados universalmente e incorporados a Internet bajo la guía del consorcio [W3C](#). Hoy en día los sitios web siguen un modelo basado en la arquitectura cliente-servidor con tres elementos comunes:

- **El lado del servidor (server-side):** incluye el hardware y software del servidor Web así como diferentes elementos de programación y tecnologías incrustadas. Po ejemplo, desarrollo en lenguaje Java o tecnologías de servidor de bases de datos que soporten múltiples sitios web.
- **El lado del cliente (client-side):** este elemento hace referencia a los navegadores web y está soportado por tecnologías como **HTML, CSS y lenguajes** como **JavaScript** y controles ActiveX, los cuales se utilizan para crear la presentación de la página o proporcionar características interactivas. Es justamente aquí dónde nos vamos a centrar a lo largo de este primer punto y el siguiente.
- **La red (Internet):** describe los diferentes elementos de conectividad (capacidad que tiene un dispositivo para poder conectarse a otros) utilizados para mostrar el sitio web al usuario.

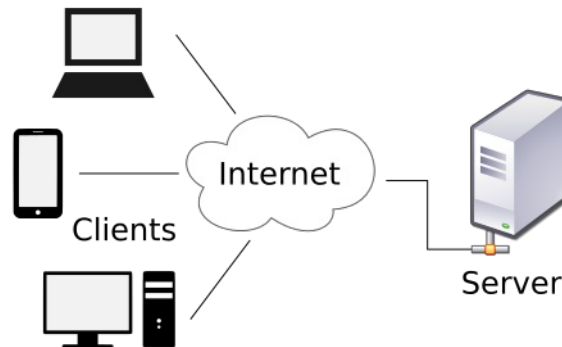


Imagen en Wikipedia de [DavidV](#) bajo licencia [Creative Common CCO](#)

## 1.1 Lenguajes de programación en entorno cliente



Hemos estudiado en el apartado anterior que hay dos grupos de tecnologías, **client-side** aquellas que son ejecutadas en el cliente, generalmente en el contexto del navegador web y cuando los programas o tecnologías son ejecutadas o interpretadas por el servidor estamos hablando de programación **server-side**. Uno de los objetivos en la programación web es saber escoger la tecnología correcta para tu trabajo ,el lenguaje más apropiado en cada caso. En general las tecnologías client-side y server-side poseen características que las hacen complementarias más que adversarias.

Por ejemplo, piensa en un *formulario web*, para su correcta funcionalidad intervienen las tecnologías del lado del cliente y el lado del servidor. El formulario nos permite recoger información y grabarla en una base de datos, es obvio que tendría más sentido chequear el formulario en el lado del cliente para asegurarnos que la información introducida es correcta, justo antes de enviar la información a la base de datos del servidor. La programación en el lado del cliente consigue que la validación del formulario sea mucho más efectiva y que el usuario se sienta menos frustrado al cubrir los datos en el formulario.

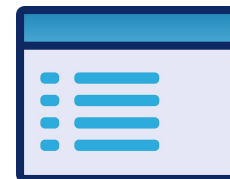


Imagen de Pixabay de  
Cleker bajo licencia Creative  
Common CCO

Entorno a los lenguajes de programación de entorno cliente, se distingue:

- Lenguajes que nos permiten dar formato y estilo a una página web (HTML, CSS, etc.).
- Lenguajes que nos permite aportar dinamismo a páginas web ( lenguajes scripting tales como JavaScript o VBScript)

En este tema nos vamos a centrar principalmente en estos últimos, los lenguajes de scripting, y en particular en el lenguaje JavaScript. Lo estudiaremos en el siguiente punto, pero debes saber que es el lenguaje más utilizado en la programación en el lado del cliente, y está soportado mayoritariamente por todas las plataformas.

La siguiente imagen presenta un esquema de las cuatro **capas del desarrollo web en el lado del cliente**, en la que se puede ver que JavaScript se sitúa en la capa superior gestionando el comportamiento de la página web.

Comportamiento (JavaScript)	Contenido Estructurado (documento HTML)
Presentación (CSS)	
Estructura (DOM / estructura HTML)	
Contenido (texto, imágenes, vídeos, etc.)	

Imagen estructuras de cuatro capas del desarrollo web en el lado del cliente de creación propia

### Características

Los lenguajes de programación para clientes web no sustituyen a la programación en el lado del servidor. Cualquier web que reaccione dinámicamente a interacciones del usuario o que almacene datos, estará gestionada por lenguajes de script en el lado del servidor, incluso aunque usemos JavaScript en el cliente para mejorar la experiencia de usuario, por varios motivos:

- Los lenguajes de script por sí mismo no pueden escribir ficheros en el servidor.
- No todos los clientes (navegadores, dispositivo móviles, etc.) web ejecutan JavaScript.

Uno de los aspectos que favorecen el auge de los script ha sido la integración con [ajax](#). Por ejemplo, así es como funciona una de las webs más populares en Internet, el servicio de [Google Maps](#). ¿Pero qué nos aporta los lenguajes en el cliente? ¿Cuál son las ventajas? Dan soluciones para:

- Conseguir que nuestra página web responda o reaccione directamente a la interacción del usuario con elementos de formulario y enlaces hipertexto.
- Controlar múltiples ventanas o marcos de navegación.
- Pre-procesar datos en el cliente antes de enviarlos al servidor.
- Modificar estilos y contenido en los navegadores de forma dinámica e instantáneamente, en respuesta a interacciones del usuario.
- Solicitar ficheros del servidor, y enviar solicitudes de lectura y escritura a los lenguajes de servidor.

**JavaScript (JS)** es un lenguaje de programación interpretado, del lado del cliente ([client-side](#)), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario haciéndolas dinámicas. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. JavaScript fue inventado por Brendan Eich en 1995 y se convirtió en un estándar ECMA en 1997. ECMAScript 6 (publicado en junio de 2015) es la última versión de JavaScript.

### ¿Por qué aprender JavaScript?

JavaScript es uno de los tres idiomas fundamentales en el desarrollo web en el cliente, como hemos estudiado:

1. HTML define el contenido de las páginas web.
2. CSS especifica el diseño de las páginas web.
3. JavaScript programar el comportamiento de las páginas web.

Hoy en día casi todas las páginas web contienen JavaScript, que se ejecuta en el navegador web del visitante. JavaScript permite crear funcionalidades específicas en las páginas web y si por alguna razón es deshabilitado, el contenido o la funcionalidad de la página puede quedar limitada o no disponible.

### Características

A continuación definimos algunas de las características más importantes de este lenguaje:

- Dinámico: evalúa expresiones como expresadas como cadenas en tiempo de ejecución
- Imperativo y estructurado: es compatible con gran parte de la [estructura de programación](#) (por ejemplo, sentencias if, bucles for, etc.)
- Funcional: hace uso de las funciones, muchas de ellas ya programadas y disponibles en librerías.

### Sintaxis y semántica: ejemplos

En este punto veremos algún ejemplo sencillo de desarrollo en JavaScript. Es importante saber que las variables en JavaScript se definen usando la palabra clave `var` y los [comentarios](#) van precedidos con 2 barras diagonales.

```
var x; // define la variable x, aunque no tiene ningún valor asignado por defecto
var y = 2; // define la variable y y le asigna el valor 2 a ella
```

Aunque desarrollar JavaScript requiere esfuerzo y tiempo, el objetivo es que al menos te familiarices con su sintaxis y conozcas algunas funciones usuales. Vamos a ver dos ejemplos, pero no olvides consultar la siguiente web: [ejemplos JavaScript](#).

Vamos a crear un botón que tras pulsarlo nos muestre automáticamente la fecha y hora actuales. Si te fijas, asociamos el evento JavaScript con el atributo `onclick` del elemento `button` de html y hacemos uso de la función predefinida `Date()`. Para saber dónde mostrar el mensaje de la fecha hemos creado un párrafo con un identificador(demo) al que llamamos desde `onclick`.

```
<!DOCTYPE html>
<html>
<body>

<h1>Mi primer JavaScript</h1>

<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Haz clic para conocer la fecha y hora.</button>

<p id="demo"></p>

</body>
</html>
```

Imagen código primer JavaScript de creación propia

## Mi primer JavaScript

Haz clic para conocer la fecha y hora..

Sat Sep 10 2016 14:16:11 GMT+0200 (Hora de verano romance)

Imagen en navegador después de hacer click de creación propia

Otro ejemplo muy usual es la creación de ventanas de aviso o alerta. Por ejemplo vamos a crear una alerta que salga tras hacer click en un botón. Se crea una función asociada al botón Pulsa con el atributo `onclick`.

```
<!DOCTYPE html>
<html>
<body>
<p>Click para obtener una caja de alerta</p>
<button onclick="myFunction()">Pulsa</button>

<script>
function myFunction() {
    alert("Esto es una alerta");
}
</script>
</body>
</html>
```

Click para obtener una caja de alerta

Pulsa

Imagen en navegador antes de hacer clic de creación propia

Imagen código función alerta de creación propia

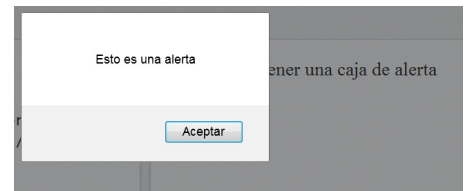


Imagen en navegador después de hacer clic de creación propia

## Para saber más



Imagen en pixabay de StartupSP bajo licencia Creative Common CCO

Actualmente, existen numerosas y variadas formas de aprender JavaScript en Internet, aquí te presentamos unos tutoriales autodidactas donde las metas y el ritmo a seguir lo marcas tú. En primer lugar la página de referencia [w3schools](#) con los ejemplos (try it yourself), y el curso de [CodeAcademy](#). Sin olvidar los video tutoriales que ofrecen los canales [khanacademy](#), [programalotu](#) y [jconde](#).

En el contexto de un sitio web, con Javascript puedes hacer todo tipo de acciones e interacciones. En un primer momento, se utilizaba para validar formularios, mostrar cajas de diálogo y poco más. Hoy es el motor de las aplicaciones más conocidas en el ámbito de Internet: Google, Facebook, Twitter, Outlook... absolutamente todas las aplicaciones que disfrutas en tu día a día en la Web tienen su núcleo realizado en JavaScript.

## 2.1 Herramientas de desarrollo



La mejor forma de aprender **JavaScript** es tecleando el código HTML y JavaScript en un simple documento de texto. Hoy en día existen múltiples **editores**, al igual que ocurre con HTML y CSS es preferible un editor sencillo pero que tenga funcionalidades entre otras: sintaxis con codificación de colores que resalte automáticamente en diferente color o tipos de letra los elementos del lenguaje tales como objetos, comentarios, funciones, variables, etc.

- Windows : Notepad++, Aptana Studio, Bluefish, Eclipse, NetBeans.
- Macintosh : Aptana Studio, Bluefish, Eclipse, KompoZer, Nvu.
- Linux : KompoZer, Amaya, Quanta Plus, Bluefish, codetech, etc.

Otro elemento fundamental es el **navegador web**, recomendadas las últimas versiones para evitar problemas de seguridad e incompatibilidades. Una recomendación muy interesante es el disponer de 2 o 3 tipos de navegadores diferentes, ya que así podrás comprobar la compatibilidad de tu página web y ver si tu código fuente de JavaScript se ejecuta correctamente en todos ellos. Otro aspecto muy importante es la **validación** que nos ahorrará mucho tiempo depurando nuestro código en la detección de errores.

### ¿Dónde se ubica el código JavaScript?

JavaScript se puede ubicar en las secciones `<body>` y `<head>` de una página HTML, o en ambos, pero siempre el código JS debe ser insertado entre las etiquetas `<script>` y `</script>`. Además es una buena práctica decidir y mantener el código siempre en el mismo lugar.

Veamos el mismo ejemplo de JavaScript en `<head>` y en `<body>` respectivamente.

```
<!DOCTYPE html>
<html>
<head>
<script>
    function myFunction() {
        document.getElementById("demo").innerHTML = "Párrafo cambiado.";
    }
</script>
</head>
<body>
    <h1>JavaScript en Head</h1>
    <p id="demo">Un párrafo.</p>
    <button type="button" onclick="myFunction()">Pulsa</button>
</body>
</html>
```

Imagen de creación propia de código JS en `<head>`

```
<!DOCTYPE html>
<html>
<body>
    <h1>JavaScript en Body</h1>
    <p id="demo">Un párrafo.</p>
    <button type="button" onclick="myFunction()">Pulsa</button>

    <script>
        function myFunction() {
            document.getElementById("demo").innerHTML = "Párrafo cambiado.";
        }
    </script>
</body>
</html>
```

Imagen de creación propia de código JS en `<body>`

Existe una tercera opción, los scripts también puede ser insertados mediante archivos externos con extensión `.js`. Este método es práctico cuando se utiliza el mismo código fuente (js) en diferentes páginas web, pues evitar redundar código y mejora los errores. La sintaxis para incluirlo sería:

```
<!DOCTYPE html>
<html>
<body>
    <script src="miScript.js"></script>
</body>
</html>
```

### 3. Usabilidad Web



#### ¿Qué es Usabilidad Web?

La Usabilidad es la medida de la calidad de la experiencia que tiene un usuario cuando interactúa con un producto o sistema.

Otra definición es la del académico *Yusef Hassan* (Universidad de Granada) al indicar que "la usabilidad es la disciplina que estudia la forma de diseñar Sitios Web para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible" y agregar que "la mejor forma de crear un Sitio Web usable es realizando un diseño centrado en el usuario, diseñando para y por el usuario, en contraposición a lo que podrá ser un diseño centrado en la tecnología o uno centrado en la creatividad u originalidad".

Si atendemos a la norma internacional **ISO/DIS 9241-11: Ergonomics of human-system interaction**, la usabilidad es el grado en que un producto o servicio puede ser empleado por usuarios específicos para conseguir objetivos concretos con efectividad, eficiencia y satisfacción, en un contexto de uso.

¿Qué crees que sucede cuando un usuario accede a una página web que no le resulta amigable? Veamos un caso práctico: pensemos en nuestra reacción cuando queremos adquirir un producto, pero el proceso de compra online es engorroso o poco confiable. Al final, es muy posible que optemos por la competencia y desechemos totalmente la primera opción para futuras transacciones.

Igual ocurre con un blog, un medio de comunicación digital o una página especializada, por eso es tan importante la usabilidad. En este punto aprenderemos cómo identificar o medir el grado así como diseñar esquemas que cumplan nuestros objetivos.

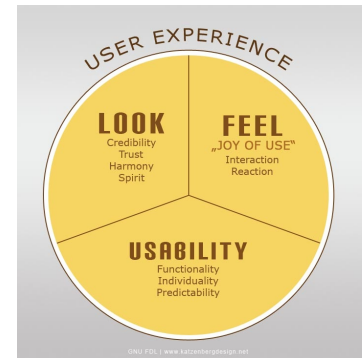


Imagen en Wikipedia de [Daniel Wurst](#) bajo licencia [Creative Common CCO](#)

#### Para saber más

##### La importancia de una buena usabilidad



Imagen en Pixabay de [fancyrave1](#) bajo licencia [Creative Common CCO](#)

Cuando diseñamos una página web no sólo es necesario considerar el posicionamiento en buscadores, sino también la usabilidad y la experiencia de usuario (UX).

En otras palabras, el consumo de productos o contenidos va en función de la usabilidad del sitio, tal como lo refleja el trabajo de [Jakob Nielsen](#), uno de los profesionales más respetados en esta materia y ampliamente conocido por su decálogo de [principios heurísticos para lograr mayor usabilidad](#).

Estos diez mandamientos que puedes consultar en el enlace anterior, muestran cómo en el diseño web se deben tener en cuenta las características y habilidades de los usuarios dado que interactúan con el entorno virtual. Por este motivo la usabilidad se vuelve un elemento vital en el diseño y desarrollo de este tipo de sistemas.



## 3.1 Interacción persona-ordenador. Interfaz de usuario.



Imagen de Pixabay de FirmBee bajo licencia Creative Common CCO

En el comienzo de la era de las tecnologías, información y comunicaciones, los ordenadores eran grandes y caros, por lo tanto los usuarios eran personas especializadas que no sabían cómo hacer más fácil su uso. Posteriormente, los costes de los ordenadores se abarataron y disminuyeron su tamaño convirtiéndose en herramientas de utilidad general. Se produjo una expansión de los ordenadores propiciada por la conectividad que ofrecía la red de redes, Internet. Como consecuencia los lenguajes de programación y los sistemas operativos avanzaron, permitiendo la creación de nuevos tipos de aplicaciones.

**La interacción persona-ordenador (IPO)** es una disciplina que se ocupa del diseño, la evaluación y la implementación de sistemas informáticos interactivos que utilizan las personas [en un contexto social y cultural] y con el estudio de los fenómenos principales relacionados con esos sistemas.

A continuación veremos cuáles son los objetivos de esta disciplina, para que los sistemas sean:

- **Útiles:** Que permitan realizar aquello para lo que se crearon.
- **Seguros:** Que permitan al usuario garantizar la seguridad del sistema.
- **Efectivos:** Que permitan realizar las tareas para las que se diseñaron sin errores.
- **Eficientes:** Que permitan ejecutar tareas satisfactoriamente y en corto espacio de tiempo.
- **Usables:** Que sean fáciles de usar y de aprender.

### Interfaz de usuario

La interfaz gráfica de usuario (del inglés *graphical user interface*) es una parte muy importante del éxito o fracaso de una aplicación. ¿Pero qué es realmente? La GUI es un conjunto de medios por los cuales las personas (los usuarios) interactúan con una determinada máquina, sistema o herramienta que incluye software, permitiéndoles su manipulación para utilizar sus funcionalidades, mediante un lenguaje de entrada y un lenguaje de salida.

Existen varios **estilos de interacción**, los más usuales son:

- **Selección por menú:** Los usuarios leen una lista de ítems, seleccionan el más apropiado para su tarea, y observan los efectos.
- **Rellenando huecos:** Los usuarios ven distintos campos, mueven el cursor entre los campos e introducen los datos cuando lo desean.
- **Lenguaje de comandos:** Los usuarios expertos se sienten cómodos con él ya que consideran que tienen el control y la iniciativa.
- **Lenguaje natural:** La interfaz reconoce el diálogo con el usuario.

La **usabilidad** determina las características que debe cumplir una **interfaz**:

- **Fáciles de aprender:** Para que el usuario pueda comenzar a trabajar rápidamente con el sistema.
- **Fáciles de recordar:** Para que los usuarios ocasionales puedan volver a utilizarlo tras un tiempo.
- **Eficientes:** Para que una vez se sepa utilizarlos se aumente la productividad.
- **Sin errores (efectivos):** Para que el usuario incurra en pocos errores durante su uso, y si lo hace, puedan recuperarse.
- **Satisfactorias:** Para que el usuario quede satisfecho.

¿Cómo conseguimos alcanzar los puntos anteriores? Se deben seguir una serie de **principios de usabilidad**:

- Diálogos sencillos.
- Utilización del lenguaje sin tecnicismos.
- Minimización de la carga de memoria del usuario.
- Consistencia, evitando el uso de duplicidades.
- Retroalimentación, dando al conocer al usuario qué ocurre.
- Salidas y resultados claros.
- Imágenes relacionadas.
- Enlaces e hipervínculos apropiados.
- Mensajes de error apropiados.
- Prevención de errores conocidos.
- Ayuda y documentación.

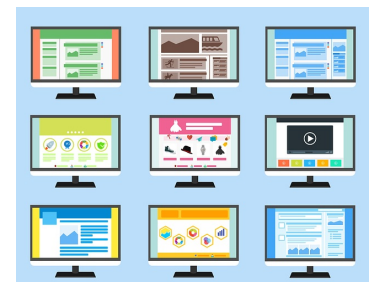


Imagen de Pixabay de JuralMin bajo licencia Creative Common CCO

## Para saber más

### Últimas tendencias claves para los diseñadores digitales



Imagen de Pixabay de JuralMin bajo licencia Creative Common CCO

En el caso del desarrollo de *software* y el diseño digital también existen tendencias que son susceptibles de un listado. A continuación se nombran algunas de las más importantes.

Las líneas más importantes seguidas en el sector son el desarrollo en la nube, interfaces de desarrollo de aplicaciones (APIs), uso especial de tipografías, crecimiento del *storytelling*, *diseño responsive* (que estudiamos en el tema de hojas de estilo), uso del *parallax*, apuesta por la sencillez son preferencias de los últimos años, imágenes grandes y experiencia multimedia (vídeos cortos que brindan una breve sinopsis de la web gracias al uso de HTML5).





### Diseño de la web y su contenido

El diseño de una web tiene como punto clave inicial el desarrollo de una *página inicial o principal* que deje suficientemente claro de qué trata el sitio. Debería responder favorablemente a cuestiones tales como, ¿Qué es esto? ¿Qué puedo hacer aquí? ¿Qué servicios/productos ofrecen? ¿Por qué debería estar aquí y no en otro lugar?

El primer impacto visual que realizamos sobre una página web es fundamental para garantizar su éxito, no obstante pasado ese momento los estudios de usabilidad indican que los usuarios centran su interés en el contenido. Aunque es importante ser gramaticalmente correcto, también es importante presentar el contenido de un modo que atraiga a los usuarios. Hay varios puntos que deben cumplirse:

- El texto debe ser correcto y conciso. El lenguaje debe ser claro y legible (tipografía, color etc).
- Facilidad en la búsqueda: utilizar párrafos cortos, subencabezados y listas con viñetas. Incluir titulares.
- Incorporar hipervínculos para dividir la información extensa en varias páginas.
- Añadir contenido multimedia, comprobando previamente que no perjudica los tiempos de respuesta en exceso.

### Técnicas de evaluación

La evaluación es la valoración de la conformidad entre el funcionamiento real del sistema desde el punto de vista de las propiedades de la usabilidad y el funcionamiento deseado. La evaluación permite detectar problemas en la interacción entre usuario y sistema:

- Evaluación formativa: Mejorar la interfaz.
- Evaluación sumativa: Elegir la mejor entre dos o más opciones.

Hay una serie de ventajas que proporciona la evaluación de la usabilidad:

- Reducción de los costes de producción.
- Reducción de los costes de mantenimiento y apoyo.
- Reducción de los costes de uso.
- Mejora en la calidad del producto.

Por otro lado están disponibles varias formas de llevar a cabo la evaluación:

- Automática: Se basan en las especificaciones de las interfaces a través de programas software.
- Empírica: Realizando pruebas sobre las interfaces utilizando usuarios reales.
- Formal: Utilizando modelos exactos y fórmulas.
- Informal: Utilizando "reglas heurísticas", conocimiento y experiencia de expertos.

## Testing



Imagen en Pixabay de [Flag](#) bajo licencia [Creative Commons CCO](#)

## Para saber más

### Pruebas de usabilidad

Estas pruebas se basan en el desarrollo de experimentos para obtener información específica acerca de un diseño. Los experimentos se realizan con usuarios reales que ejecutan tareas reales. ¿Qué conseguimos? Nos permite obtener medidas cuantitativas de la usabilidad (tiempo que se tarda en ejecutar una tarea), identificar errores (número, causa, etc.) así como el grado de satisfacción del usuario.

¿Cómo se lleva a cabo?

- a. Definir qué queremos estudiar
- b. Diseñar la prueba
- c. Seleccionar a los usuarios
- d. Configurar el test
- e. Ejecutar el experimento
- f. Obtener información del usuario
- g. Analizar los datos

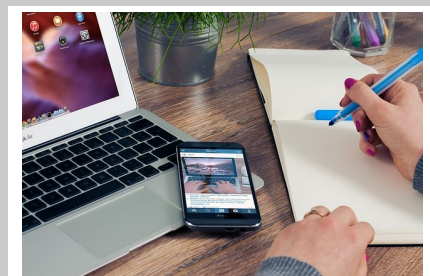


Imagen en Pixabay de [FirmBee](#) bajo licencia [Creative Commons CCO](#)

Finalmente, existe una serie de herramientas para medir la efectividad de los componentes que determinan cuán usable es un sitio web. Por ejemplo, [CrazyEgg](#), [Google Analytics](#) y [ClickTale](#), entre otras. La usabilidad podría incrementar el tiempo que los usuarios pasan en la página y convertir a dicho usuario en un distribuidor de valiosa información.

¿Qué entendemos por **accesibilidad**?

Es la condición que deben cumplir los entornos, productos y servicios para que sean comprensibles, utilizables y practicables por todas las personas" según lo definido en la Ley 51/2003, de 2 de diciembre, de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad.

### Accesibilidad Web

Si el concepto anterior lo aplicamos a los entornos Web, podemos definir **la accesibilidad web** como la capacidad de que todas y todos los usuarios de Internet puedan acceder a estos recursos. Es la posibilidad de que un **sitio o servicio Web** sea visitado y utilizado de forma satisfactoria por el mayor número de **personas**, independientemente de las **limitaciones** que éstas tengan o las derivadas de su **entorno**". (*Instituto Nacional de Tecnologías de la Comunicación (INTECO), España*).

Veamos un ejemplo de contenido no accesible para usuarios con discapacidad visual, como el daltonismo:





CURSO	CURSO
<ul style="list-style-type: none"><li>• Si pulsa el círculo verde comenzará el curso</li><li>• Si pulsa el círculo rojo se eliminará el curso</li></ul>	<ul style="list-style-type: none"><li>■ Si pulsa el círculo verde comenzará el curso</li><li>■ Si pulsa el círculo rojo se eliminará el curso</li></ul>
 	 
	Eliminar Comenzar
No accesible ☹	Accesible ☺

Imagen de creación propia

### Estándar

El concepto de “estándar” se puede definir como la disposición destinada a usos comunes y repetidos, con el fin de obtener un nivel de ordenamiento óptimo en un contexto, con el objetivo de crear un lenguaje común. Se utiliza el término “estándar” en general para referirse a diferentes disposiciones: “Norma”, “Especificación” y “Recomendación”. Teniendo en cuenta que en unos casos se trata de estándares “**de iure**” (oficiales: ISO, UNE) y en otros de estándares “**de facto**” (W3C, ETSI).

### Estándares sobre accesibilidad web

Estos estándares ofrecen un **marco común** para regular diferentes aspectos relacionados con el desarrollo de sistemas web accesibles y la evaluación del nivel de accesibilidad. El **principal objetivo** de estos estándares es **mejorar la accesibilidad** de los productos web que se ponen a disposición de los usuarios a través de protocolos de Internet, mediante un navegador web.

La W3C (World Wide Web Consortium) define entorno a la regulación de la accesibilidad del contenido web la normativa **W3C/WAI** (Web Accessibility Initiative) y **WCAG 2.0: Web Content Accessibility Guidelines**.

También existe legislación sobre la accesibilidad web impuesta por distintos países y a nivel de organismos como la unión europea. Por ejemplo en España se define el **Real Decreto 1494/2007, de 12 de noviembre, por el que se aprueba el Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a la sociedad de la información**.



Imagen de creación propia con iconos CC obtenidos de Wikipedia

### Análisis de la accesibilidad de contenidos Web

**WCAG 2.0: Web Content Accessibility Guidelines** establece 61 requisitos para las páginas Web basados en cuatro principios básicos:

1. **Perceptible:** La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que ellos puedan percibirlos.
2. **Operable:** Los componentes de la interfaz de usuario y la navegación deben ser operables.
3. **Comprensible:** La información y el manejo de la interfaz de usuario deben ser comprensibles.
4. **Robusto:** El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de aplicaciones de usuario, incluyendo las ayudas técnicas.

Estos principios básicos (1..4) se descomponen en 12 pautas ( 1.1, 1.2 .... 4.1..) y las pautas se descomponen en 61 requisitos o criterios de conformidad (1.1.1, ..3.1.2, ...).



La accesibilidad **beneficia a las personas con discapacidades** visuales, auditivas, físicas o motrices. Pero también beneficia a otros grupos como son los usuarios de avanzada edad, usuarios que no dominan el idioma o aquellos inexpertos.

Imagen de W3C bajo licencia Creative Common CCO

## Para saber más

### Ejemplo de criterio WCAG 2.0

Criterio de conformidad “PERCEPTIBLE” ( 1.1.1 Contenido no textual) define que:

“Todo contenido no textual que se presenta al usuario tiene una alternativa textual que cumple el mismo propósito”. Este elemento ya lo estudiamos en HTML debemos utilizar el atributo alt, en caso de que no se pueda visualizar la imagen aparecerá el texto contenido en este atributo alt dando información al usuario.

``

Sitúa el ratón sobre el logo de google chrome que está a la derecha, verás como se muestra el alt anteriormente descrito.



## 4.2 Evaluación

En la evaluación de la accesibilidad de sitios web se hace uso de una serie de herramientas de evaluación automática. Algunas de las más importantes son:

- Validadores de gramática (HTML, CSS)
- Validadores de puntos de control de accesibilidad (WCAG)
- Barras de herramientas y extensiones de navegadores Web
- Evaluadores de color y contraste
- Simuladores de discapacidades
- Navegadores de texto
- Productos de apoyo

Hoy en día existen numerosas páginas web que ofrecen la evaluación de accesibilidad WCAG 2.0 de páginas Web de forma online: [TAW](#), [Tingun](#) y [Wave](#). Así como otras utilidades: [Validadores](#), Barras de herramientas en navegadores ([Mozilla Firefox](#), [Google Chrome](#)).



The screenshot shows the TAW (TIC Centro Tecnológico) accessibility evaluation results page. The page has a teal header with the TAW logo and navigation tabs: Resumen, Vista Marcada, Detalle, and Listado. The main content area is titled "Resumen de resultados" and "Información del análisis". It displays the following information:

- Recurso:** <http://www.juntadeandalucia.es/educacion/portals/web/ieda>
- Fecha:** 05/09/2016 12:42
- Pautas:** WCAG 2.0
- Nivel del análisis:** AA
- Tecnologías:** HTML, CSS

The results are summarized in three categories:

- 33 Problemas** (en 3 criterios de éxito): Son necesarias correcciones. Perceptible 10, Operable 0, Comprensible 0, Robusto 23.
- 1.407 Advertencias** (en 9 criterios de éxito): Es necesario revisar manualmente. Perceptible 14, Operable 3, Comprensible 12, Robusto 1378.
- 18 No verificados** (en 18 criterios de éxito): Comprobación completamente manual. Perceptible 4, Operable 9, Comprensible 5, Robusto 0.

Imagen de creación propia evaluación taw de la página web IEDA

## 5. Para ampliar



Tal y como vimos en el tema de HTML&CSS, existen múltiples páginas y plataformas que nos ayudan con el aprendizaje, en este caso JavaScript. Puedes consultar más información al respecto en [w3schools](https://www.w3schools.com).

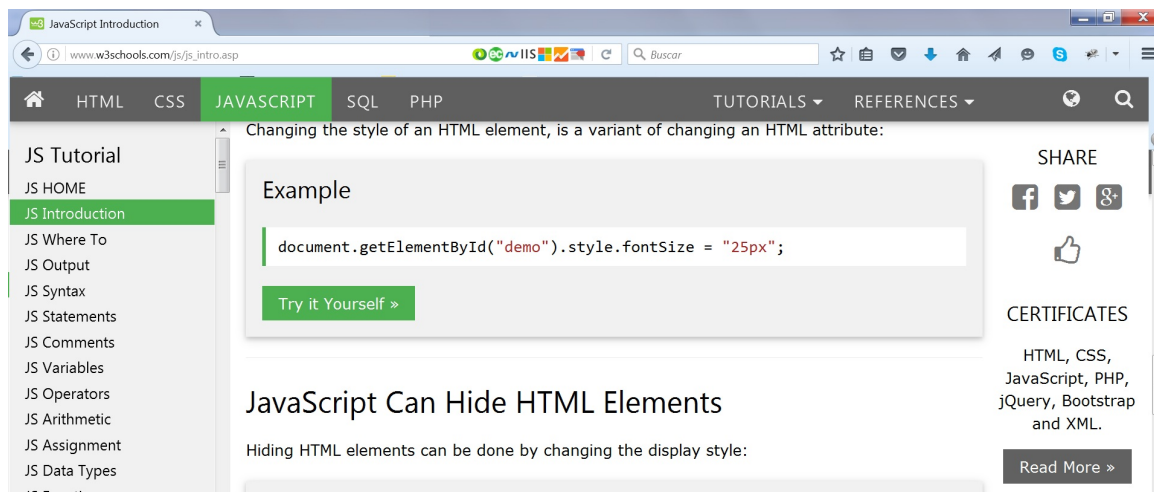


Imagen de web weschool js de obtención propia

Ya sabemos que JavaScript nos permite añadir dinamismo al web a través de funciones, pero ¿debo crear yo las funciones? La respuesta es no, existen las llamadas bibliotecas (library) con las funciones ya implementadas, ¡solo debemos hacer uso de ellas "llamarlas"! Entonces ya podemos hablar de jQuery (la biblioteca de JavaScript más utilizada).

**jQuery** es una [biblioteca](#) de [JavaScript](#), creada inicialmente por [John Resig](#), que permite simplificar la manera de interactuar con los documentos [HTML](#), manipular el árbol [DOM](#), manejar eventos, desarrollar animaciones y agregar interacción con la técnica [AJAX](#) a páginas web. Es [software libre y de código abierto](#), la gran ventaja que nos proporciona es que a través de un único fichero JavaScript, contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. Esto nos permite cambiar el contenido de una página web sin necesidad de recargarla. Para ello utiliza las funciones `$()` o `jQuery()`.

Esta librería se integra a la perfección con HTML5 y CSS3, en los diseños responsive para distintas plataformas. Encontrarás múltiples recursos para aprender jQuery en la web, consulta estas [demos](#) para hacerte una idea.



Imagen de pixabay de [openrb](#) bajo licencia de [Creative Common CCO](#)



## Comprueba lo aprendido

¿Qué etiqueta nos permite introducir el código JavaScript?

☐ <javascript>

☐ <script>

☐ <js>

### Solution

1. Incorrecto
2. Correcto
3. Correcto

¿Cuál es el lugar correcto para insertar código JavaScript?

☐ En la sección <head>

☐ En la sección <body>

☐ En ambas secciones es correcto <head> y <body>

### Solution

1. Incorrecto
2. Incorrecto
3. Correcto

¿Cuál es la sintaxis correcta para vincular el código HTML con una fichero JavaScript externo?

☐ <script href="xxx.js">

☐ <script src="xxx.js">

☐ <script name="xxx.js">

### Solution

1. Incorrecto
2. Correcto
3. Incorrecto

¿Existen herramientas que validan la web , para gestionar la....?

☐ Usabilidad

☐ Accesibilidad

☐ Ambas

### Solution

1. Incorrecto
2. Correcto
3. Incorrecto

¿Cuántos niveles de accesibilidad hay a día de hoy?

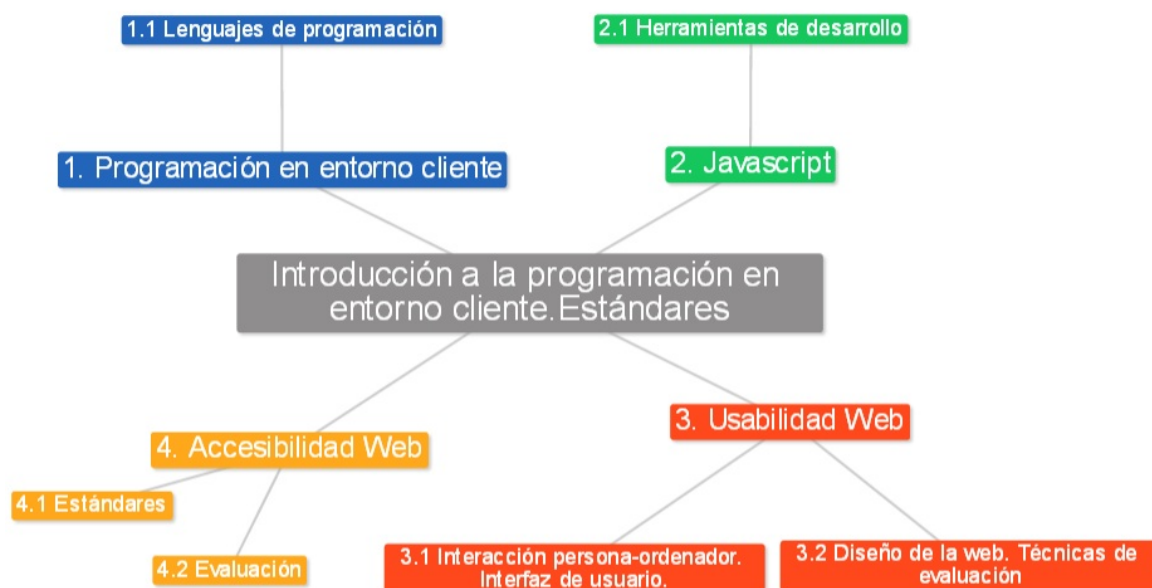
☐ 3

☐ 2

☐ 1

**Solution**

1. [Correcto](#)
2. [Incorrecto](#)
3. [Incorrecto](#)



## Aviso legal

El presente texto (en adelante, el "**Aviso Legal**") regula el acceso y el uso de los contenidos desde los que se enlaza. La utilización de estos contenidos atribuye la condición de usuario del mismo (en adelante, el "**Usuario**") e implica la aceptación plena y sin reservas de todas y cada una de las disposiciones incluidas en este Aviso Legal publicado en el momento de acceso al sitio web. Tal y como se explica más adelante, la autoría de estos materiales corresponde a un trabajo de la **Comunidad Autónoma Andaluza, Consejería de Educación, Cultura y Deporte (en adelante Consejería de Educación, Cultura y Deporte Andaluza )**.

Con el fin de mejorar las prestaciones de los contenidos ofrecidos, la Consejería de Educación, Cultura y Deporte Andaluza se reservan el derecho, en cualquier momento, de forma unilateral y sin previa notificación al usuario, a modificar, ampliar o suspender temporalmente la presentación, configuración, especificaciones técnicas y servicios del sitio web que da soporte a los contenidos educativos objeto del presente Aviso Legal. En consecuencia, se recomienda al Usuario que lea atentamente el presente Aviso Legal en el momento que acceda al referido sitio web, ya que dicho Aviso puede ser modificado en cualquier momento, de conformidad con lo expuesto anteriormente.

### **1. Régimen de Propiedad Intelectual e Industrial sobre los contenidos del sitio web**

#### **1.1. Imagen corporativa**

Todas las marcas, logotipos o signos distintivos de cualquier clase, relacionados con la imagen corporativa de la Consejería de Educación, Cultura y Deporte Andaluza que ofrece el contenido, son propiedad de la misma y se distribuyen de forma particular según las especificaciones propias establecidas por la normativa existente al efecto.

#### **1.2. Contenidos de producción propia**



