

DESARROLLO DE WEB ENTORNO CLIENTE  
TÉCNICO EN DESARROLLO DE APLICACIONES WEB

# Lenguajes y herramientas de programación en clientes web

---

02

# ÍNDICE

<b>/ 1. Introducción y contextualización práctica</b>	<b>3</b>
<b>/ 2. Características de los lenguajes de programación en clientes web</b>	<b>4</b>
2.1. Motor de ejecución en navegadores web	4
2.2. Principales navegadores web	5
<b>/ 3. Desarrollo en entornos web de cliente</b>	<b>5</b>
<b>/ 4. Caso práctico 1: “Viewport”</b>	<b>6</b>
<b>/ 5. Entornos de programación más utilizados</b>	<b>6</b>
<b>/ 6. Lenguajes más utilizados en clientes web</b>	<b>7</b>
6.1. Frameworks más utilizados en desarrollo en cliente	7
<b>/ 7. Integración de código mediante etiquetas</b>	<b>8</b>
<b>/ 8. Lenguajes de script en entornos web</b>	<b>8</b>
<b>/ 9. Caso práctico 2: “Aplicación bancaria”</b>	<b>9</b>
<b>/ 10. Resumen y resolución del caso práctico de la unidad</b>	<b>10</b>
<b>/ 11. Bibliografía</b>	<b>10</b>

# OBJETIVOS

*Identificar las capacidades de los distintos navegadores web.*

*Entender los mecanismos de ejecución de los navegadores.*

*Identificar los principales lenguajes de programación en entorno cliente.*

*Conocer las peculiaridades de los lenguajes de marcado para el desarrollo web.*

*Conocer las principales herramientas de desarrollo para entornos web en el cliente.*

## / 1. Introducción y contextualización práctica

El desarrollo web en cliente es un campo en constante evolución, pues el número de tecnologías está en continuo cambio. Desde los lenguajes de marcado a los lenguajes de scripting, en este contexto se introducen diferentes enfoques de programación frente al paradigma tradicional. En este tema presentaremos las principales consideraciones para el desarrollo de soluciones web en el lado del cliente.

Escucha el siguiente audio que describe el caso práctico que iremos resolviendo a lo largo de la unidad:

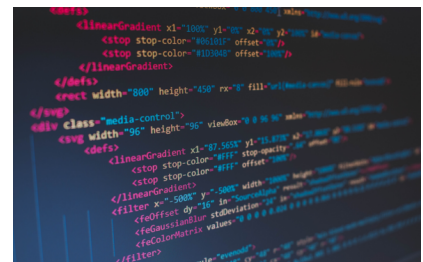


Fig. 1. Código HTML para el front-end



Audio Intro. "HTML: contenedores"

<https://bit.ly/2Z9BtOX>





## / 2. Características de los lenguajes de programación en clientes web

Actualmente, se espera que las aplicaciones web ofrezcan al usuario una alta interacción. Por ejemplo, se espera que el sistema se encuentre disponible constantemente (24/7) y accesible, asimismo, desde cualquier parte del mundo. Además, existe un amplio número de tipos de dispositivos a través de los cuales el usuario podría acceder a la aplicación web. Como es de esperar, el punto de entrada del sistema será siempre la interfaz gráfica, es decir, lo que mostremos en el navegador. Conocer cómo funciona el navegador, qué tareas realiza y cómo las realiza es fundamental para poder construir un front-end que se adapte a las necesidades de los usuarios en cada momento.

Para poder desarrollar en el cliente, debemos disponer de la interpretación que el navegador web pueda realizar del código recibido del servidor. Cuando el usuario realiza una petición web, el resultado viaja desde el servidor hasta el navegador a través del protocolo HTTP. Este se encarga de realizar la visualización del código recibido. En ocasiones sucede que diferentes navegadores realizan interpretaciones distintas del código recibido. Este funcionamiento genera problemas reales en los programadores de front-end, pues hay que realizar adaptaciones para diferentes navegadores.



Fig. 2. Alta disponibilidad web

### 2.1. Motor de ejecución en navegadores web

En la actualidad contamos con cinco principales navegadores para computadores: Chrome, IE/Edge, Firefox, Safari y Opera. En dispositivos móviles, la mayor parte de los navegadores disponibles se basan en el framework WebKit.

Todos los navegadores se dividen, como mínimo, en los siguientes componentes principales: interfaz gráfica, motor de navegación, motor de visualización, intérprete JavaScript, manejador de red y gestor de almacenamiento. De los componentes anteriores, prestaremos especial atención al motor de visualización, también conocido como layout engine. Este componente se encarga de realizar las transformaciones necesarias para poder visualizar el código HTML recibido desde el servidor y convertirlo en una página web con la que el usuario pueda trabajar.

El aspecto visual de cualquier página web dependerá de los estilos utilizados a través de hojas de estilos en cascada (CSS). A través de ellas, se pueden modificar múltiples puntos como el tamaño de la fuente, los colores, el posicionamiento, las formas, etc. El motor del navegador deberá interpretar tanto el HTML recibido, como las reglas de estilo para generar una representación válida. Se puede deducir que, dependiendo de la interpretación realizada, el resultado visual puede cambiar entre los diferentes navegadores.



Fig. 3. Motor de navegador web



Audio 1. "HTTP: navegador y servidor"  
<https://bit.ly/31THUaq>





## 2.2. Principales navegadores web

Cada uno de los principales navegadores dispone de un motor de visualización:

- **Chrome.** Inicialmente se basaba en WebKit. Sin embargo, actualmente utiliza un motor propio llamado Blink.
- **Firefox.** Se basa en el motor Gecko.
- **Safari.** Su motor se basa en WebKit.
- **Edge.** Posee su propio motor de visualización llamado BLINK.

## / 3. Desarrollo en entornos web de cliente

El desarrollo web en cliente o desarrollo en front-end se encarga principalmente de gestionar la interacción con el usuario, pues se convierte en la interfaz gráfica que, en muchas ocasiones, funciona como la única puerta de entrada a una aplicación web.

Cuando hablamos de tecnología front-end nos referimos a todas aquellas tecnologías que se ejecutan siempre en el dispositivo del usuario, es decir, en su navegador web.

El desarrollador de front-end debe ser capaz de realizar las peticiones oportunas al servidor (back-end) utilizando soluciones síncronas o asíncronas, según las necesidades de cada contexto. En la actualidad, el desarrollo en front-end se puede convertir en una tarea tediosa, pues, por un lado, el número de tecnologías existentes aumenta constantemente. Por otro lado, el número de navegadores y los tipos de dispositivos aumentan de la misma forma.

Teniendo en cuenta que el uso de dispositivos móviles aumenta continuamente, el desarrollo en cliente se vuelve más exigente. En este sentido, es necesario crear diferentes formas de visualización que contemplen diversos tamaños de pantalla.



Video 1. "Navegador, ofuscación y estándar"

<https://bit.ly/3gAoxHF>





## / 4. Caso práctico 1: “Viewport”

**Planteamiento.** Nuestro cliente nos encarga el desarrollo de un front-end que pueda visualizarse en pantallas que tengan diferente tamaño como móviles, tablets y computadores de sobremesa.

**Nudo.** ¿Qué aspectos debemos tener en cuenta para garantizar el correcto funcionamiento de nuestra web? ¿En qué navegador deberíamos centrarnos en primer lugar?

**Desenlace.** Para garantizar que el front-end se pueda visualizar correctamente, es necesario considerar la densidad de puntos por pulgada de las pantallas (dpi). A partir de ahí, debemos definir el viewport y la escala para que nos podamos ajustar al tamaño de cada dispositivo. El punto de partida sería el navegador más utilizado, Chrome. Una vez garanticemos la visualización correcta en este navegador, podremos seguir con el siguiente más utilizado, y así sucesivamente. Es importante que comuniquemos a nuestro cliente que el precio del producto será diferente dependiendo del número de navegadores que soporte nuestro front-end.



Fig. 4. Viewport en la web puede cambiar la experiencia de usuario

## / 5. Entornos de programación más utilizados

A la hora de desarrollar el front-end de una aplicación web, es necesario disponer de ciertas herramientas. Sin embargo, al contrario de lo que ocurre con la tecnología back-end, las herramientas que se utilizan en front-end no requieren de grandes recursos hardware.

En las tareas de desarrollo, normalmente se utilizan IDEs (Integrated Development Environment) para poder facilitar todas las tareas involucradas. En el desarrollo web en cliente, no es necesario utilizar este tipo de herramientas, ya que sería suficiente con editores de texto plano, lo que se traduce en un menor consumo de recursos hardware. Algunos de los editores más utilizados en la actualidad son los siguientes:

- **Sublime Text.** Es uno de los editores de texto más utilizados. Incorpora soporte para el reconocimiento de múltiples lenguajes (no solo de cliente). Está disponible para los principales sistemas operativos de propósito general.
- **Notepad++.** Es un editor de código abierto que reconoce la sintaxis de múltiples lenguajes (actualmente 27).
- **Vim.** Se trata de un editor de texto de propósito general disponible para sistemas UNIX.
- **Visual Studio Code.** Se trata de un editor de código abierto desarrollado por Microsoft. Se encuentra disponible para los sistemas operativos de propósito general. Además, ofrece una gran cantidad de complementos para realizar tareas más allá de las de un simple editor.



Fig. 5. A mayor compatibilidad, mayor es el precio del front-end.



## / 6. Lenguajes más utilizados en clientes web

Como se ha comentado anteriormente, el número de tecnologías para el desarrollo en front-end aumenta constantemente. Sin embargo, todas estas tecnologías se suelen basar en pilares similares. Los lenguajes más utilizados en el desarrollo en front-end son los siguientes:

- **HTML (Hyper Text Markup Language).** Se trata de un lenguaje de marcado estándar para la creación de páginas web. Con este lenguaje se puede definir la estructura de una página web, entendiendo por estructura la distribución del contenido.
- **CSS (Cascade Style Sheet).** Se trata de hojas de estilo en cascada que permiten cambiar la representación visual de la estructura HTML creada previamente.
- **JavaScript.** Es un lenguaje sencillo y ligero que el navegador interpreta para permitir la interacción y el control de componentes web. Se puede utilizar en contextos diferentes a entornos web. Con la evolución de este lenguaje surge TypeScript, un lenguaje de alto nivel basado en JavaScript.

Otros lenguajes involucrados, ya no para el desarrollo, sino para el intercambio de datos son estos:

- **JSON (JavaScript Object Notation).** Se trata de un formato sencillo para facilitar el intercambio de datos que puede interpretarse directamente a través de JavaScript.
- **XML (Extensible Markup Language).** Se trata de un lenguaje de marcado (basado en SGML) para el intercambio de datos. Permite representar datos de manera estructurada para permitir su intercambio en la red. Existe un gran conjunto de tecnologías que rodea a esta representación, por ejemplo: XPath, XQuery, XSLT, etc.



Fig. 6. En el desarrollo web se unen diferentes tecnologías

### 6.1. Frameworks más utilizados en desarrollo en cliente

Un framework ofrece un conjunto de funcionalidades que permite realizar determinadas tareas de una manera más sencilla. Para el desarrollo en el front-end existe un gran número de frameworks que, en ocasiones, marcan una forma de desarrollo propia.

Actualmente, algunos de los frameworks más utilizados son:

- **React.** Sigue un enfoque de desarrollo reactivo y basado en JavaScript. React fue desarrollado por Facebook para crear interfaces de una manera más sencilla e intuitiva.
- **Angular.** Ofrece diferentes mecanismos para crear, no solo interfaces gráficas, sino aplicaciones completas para entornos de escritorio y móvil.
- **Vue.js.** Permite crear interfaces de usuario de una manera sencilla y progresiva.

## / 7. Integración de código mediante etiquetas

Para poder crear un front-end correctamente es necesario trabajar, en primer lugar, con HTML. A través de este lenguaje basado en etiquetas podremos crear la distribución del contenido en la web. El lenguaje está formado por un conjunto de etiquetas que siempre deben abrirse y cerrarse.

Véase el siguiente ejemplo:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Este fragmento de código permite distinguir diferentes partes en un documento HTML:

- **Cabecera.** Está delimitado por el tag (etiqueta) <head>. En la cabecera se puede incluir metadatos y enlaces a dependencias externas.
- **Cuerpo.** Está delimitado por el tag <body>. En el cuerpo se incluye todo el contenido de la web.



Video 2. "Creación documento HTML"  
<https://bit.ly/2ZRMGTv>



## / 8. Lenguajes de script en entornos web

En los orígenes de la web, no había prácticamente interacción del usuario con la web. Es decir, el front-end solo permitía que el usuario consultase los datos visualizados. Una de las principales características de la web 2.0 es el aumento de la capacidad de interacción en el front-end, con interfaces web más potentes que se asemejan a las aplicaciones de escritorio.

El aumento de la interacción de usuario ha sido posible gracias al motor de JavaScript incluido en los navegadores. Entre las tareas que nos permite realizar JavaScript, encontramos:

- Aumento de la interacción del usuario.
- Realizar comprobaciones antes de enviar información al servidor.
- Funcionamiento asíncrono mientras el usuario realiza otras tareas.

Al crear un documento HTML, podemos incluir código JavaScript a través de la etiqueta <script>. El intérprete de JavaScript será el encargado de procesar el contenido de esta etiqueta, liberando al motor de renderizado web.





En el siguiente código se muestra un fragmento código HTML que incluye una porción de código en JavaScript.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Alert</h2>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  alert("Hola Mundo!!!");
}
</script>
</body>
</html>
```

El código anterior muestra un ejemplo en el que se incluye un botón y, al pulsarlo, se ejecuta un código JavaScript que muestra un mensaje por pantalla. Puedes copiar este ejemplo en un archivo de texto y guardarlo con extensión .html. Si lo abres en cualquier navegador, podrás observar el funcionamiento. ¡Pruébalo!

## / 9. Caso práctico 2: “Aplicación bancaria”

**Planteamiento.** Como responsable de un sistema informático del front-end de una aplicación bancaria, observas que el sistema no está informando de los posibles errores de la aplicación. Consultando los logs del navegador, nuestro sistema no muestra nada.

**Nudo.** ¿Cómo crees que se puede solucionar? Propón un código de ejemplo que muestre los logs en la consola del navegador.

**Desenlace.** En primer lugar, es necesario localizar la consola del navegador que estemos utilizando. En esta resolución utilizaremos el inspector de Chrome (es el navegador más utilizado) y buscamos la pestaña de consola. En segundo lugar, creamos un documento HTML con una sintaxis básica. Por último, incluimos un código JavaScript para mostrar los logs. A continuación, se plantea una posible solución:

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Alert</h2>
<script>
console.log('Mi primer log');
</script>
</body>
</html>
```

Como se muestra en el código anterior, para mostrar la información de log en JavaScript, es necesario utilizar llamadas a `console.log()`.

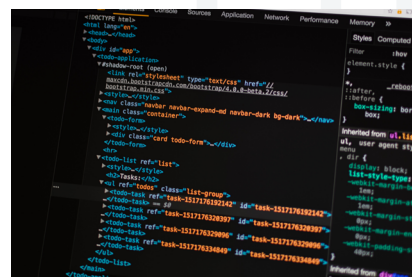


Fig. 7. Importancia de JavaScript en el desarrollo web



## / 10. Resumen y resolución del caso práctico de la unidad

A lo largo de este tema hemos realizado una introducción al desarrollo web en entornos de cliente. Para ello se han descrito las principales tecnologías de desarrollo, así como las herramientas más utilizadas en la actualidad. Hemos podido observar que existen diferentes entornos para poder desarrollar el cliente. Como norma general, tendremos que utilizar aquel entorno que nos fije nuestra empresa.

En este tema también se hemos presentado el lenguaje HTML para poder crear documentos web que permitan organizar correctamente el contenido del documento. Además, hemos estudiado qué es un lenguaje de scripting para mejorar la interacción con el usuario. Como hemos podido observar, el lenguaje HTML se ha convertido en la base para poder introducir otros lenguajes como JavaScript. A través de JavaScript podemos mejorar la interacción del usuario con el resto el sistema.

### Resolución del caso práctico de la unidad

Generalmente, en los lenguajes de programación intentamos reutilizar la mayor cantidad de código posible. En el entorno web también podemos realizar algo parecido, ya sea en JavaScript o en HTML. En el caso de HTML, la mejor forma para reutilizar código es utilizando contenedores. Un contenedor HTML no influye en la representación visual.

Además, lo aceptan todos los navegadores. La etiqueta asociada a un contenedor es la etiqueta `<div>`. Esta etiqueta pertenece al HTML desde sus orígenes. Dicho esto, para poder reutilizar el panel de redes sociales podríamos introducir todo su funcionamiento en un contenedor.

## / 11. Bibliografía

Duckett, J. & Duckett, J. (2014). *Web design with HTML, CSS, JavaScript and jQuery*. Hoboken: John Wiley & Sons.

Gómez, M. (2017). *Curso de Desarrollo Web: HTML, CSS y JavaScript*. Madrid: Anaya Multimedia.