



# **CAPITULO 1: INTRODUCCIÓN DESENVOLVEMENTO WEB EN**

## **CONTORNO CLIENTE**

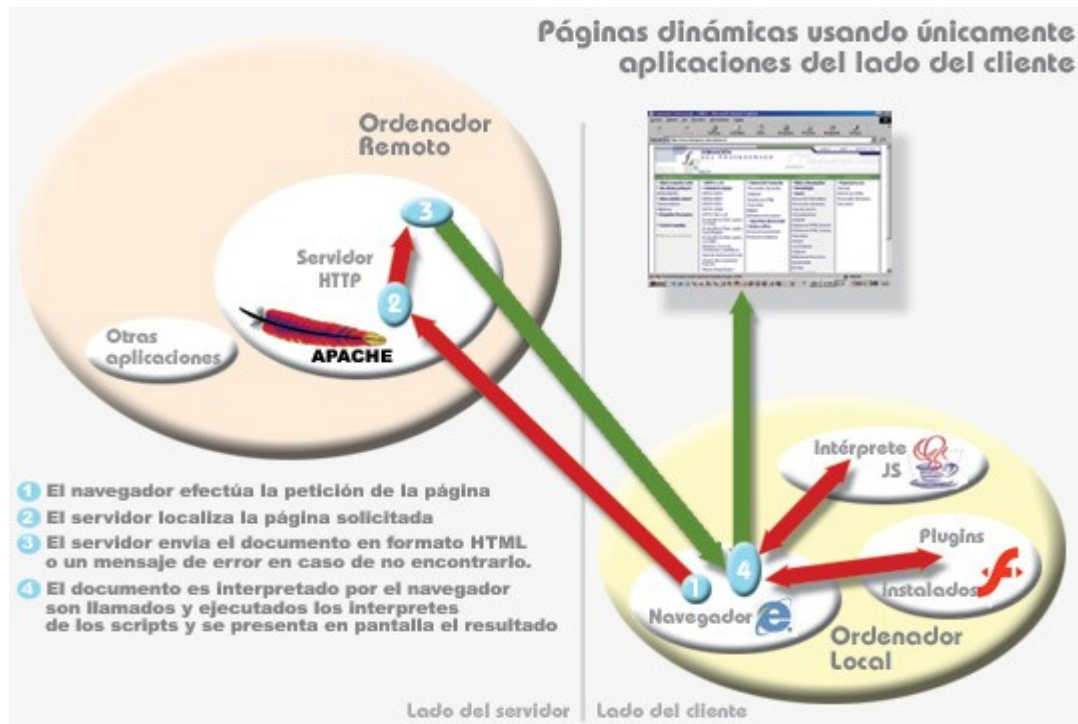
### **1 INTRODUCCIÓN AL WEB:**

La web funciona según un principio realmente sencillo: los servidores web albergan los contenidos y los clientes los solicitan por HTTP o FTP, entendiendo por clientes a los navegadores (Mozilla Firefox o Google Chrome) instalados en el sistema del usuario, donde se ejecutan. Por el contrario, los servidores web, como Apache o NGINX, forman parte de los proyectos de desarrollo web, se instalan y se ejecutan en este entorno y permiten a los clientes el acceso a los contenidos.

Los contenidos estáticos, como los clásicos elementos de HTML o las imágenes, sencillamente se envían al navegador y allí se visualizan, los contenidos dinámicos, como una Wiki, un menú desplegable o cualquier tipo de aplicación web solo funcionan por medio de scripts, los cuales se han de ejecutar e interpretar con el lenguaje de programación web correspondiente en el lado del servidor o en el lado del cliente. Es por esto que se diferencia fundamentalmente entre los lenguajes de programación del lado servidor y los lenguajes del lado cliente.

Los lenguajes de programación del lado cliente también se utilizan para realizar proyectos con contenidos dinámicos, pero, a diferencia de los lenguajes del lado servidor, no es el servidor el que ejecuta y procesa los scripts, sino el cliente solicitante. Con esta finalidad, los scripts se incluyen en el documento HTML o XHTML o se escribe en un archivo separado que se enlaza al documento principal.

Cuando un usuario solicita una página web o una aplicación web con un script de este tipo, el servidor web envía el documento HTML y el script al navegador, quien lo ejecuta y presenta el resultado final. Asimismo, los scripts del lado cliente contienen instrucciones concretas para el navegador web al respecto de cómo ha de reaccionar a ciertas acciones llevadas a cabo por el usuario como, por ejemplo, un clic en un botón específico. A menudo, el cliente ha de establecer para ello otro contacto con el servidor web.



Al ejecutarse en el navegador, el usuario puede ver el código fuente, a diferencia de lo que ocurre con los scripts del lado servidor. Como contrapartida, la interpretación de los scripts se basa en la suposición de que el navegador web entiende el lenguaje de programación web correspondiente. Como las ventanas emergentes y las herramientas de seguimiento web también se basan en el lenguaje del lado cliente y estos scripts influyen en los tiempos de carga, existen diversas extensiones para el navegador muy populares que bloquean estos scripts.

El lenguaje del lado cliente más significativo es JavaScript. Su uso se extendió rápidamente, convirtiéndose pronto en el lenguaje de programación web universal de todos los navegadores web relevantes. En los primeros tiempos de la web, Microsoft Silverlight y los Applets de Java también eran muy populares.

En teoría, cualquier lenguaje de programación web cabría utilizarse para programar del lado cliente, aunque ello implicaría que los desarrolladores de todos los navegadores más importantes deberían ponerse de acuerdo en soportarlos. No obstante, también existen soluciones alternativas que permiten escribir del lado cliente con otros lenguajes de programación web. Este es el caso de transcompiladores TypeScript, que interpretan el código y lo ejecutan como JavaScript.

## 2 PAGINAS WEB:

Se denomina **páginas Web** a documentos que contienen elementos multimedia (imágenes, texto, audio, vídeo, etc.) además de enlaces de hipertexto. Entrando en la Web a través de cualquier servidor se puede navegar por toda la red, gracias a que la mayoría de las páginas Web contienen



enlaces a otras páginas Web que pueden estar localizadas en el mismo servidor o en cualquier otro servidor de Internet. El servicio WWW es el servicio multimedia de Internet más extendido

## 2.1 PAGINAS WEB ESTÁTICAS

Son ideales para las empresas que no quieren muchas pretensiones con su sitio web, sino simplemente informar a sus clientes de sus productos y dar a conocer su perfil de empresa entre otros. La principal ventaja es lo económico que pueden resultar. Su diseño y desarrollo es más rápido que el de una web dinámica pero el usuario no puede seleccionar, ordenar o modificar los contenidos o el diseño de la página. Sus características más relevantes son:

- Ausencia de movimiento y funcionalidades.
- Realizadas en XHTML o HTML.
- Acceder al servidor para cambiar contenidos de la página.
- Opacidad a los deseos o búsquedas del visitante a la página.
- Lento y manual el proceso de actualización.

## 2.2 PÁGINAS WEB DINÁMICAS

Estas permiten la creación de aplicaciones dentro de la propia web y ofrecen una mayor interactividad con los usuarios que la visiten. Su creación es más compleja, ya que se requiere de conocimientos específicos de lenguajes de programación y gestión de bases de datos. Con este tipo de páginas web se puede hacer todo lo que se desee. Las características son:

- Infinitas posibilidades en su diseño y desarrollo.
- El visitante puede alterar el diseño, contenidos o presentación.
- Se utilizan varias técnicas de programación.
- El proceso de actualización es muy sencillo.
- Permite muchas funcionalidades como bases de datos, foros, etc.

## 3 HERRAMIENTAS UTILIZADAS

### 3.1 VSCODE: Características de Vscod

Entre las principales características de VSCode se pueden mencionar:

- **IntelliSense:** por defecto proporciona terminaciones inteligentes basadas en tipos de variables, definiciones de funciones y módulos importados.
- **Run and Debug:** es posible realizar depuración con puntos de interrupción.
- **Git integrado:** nativamente viene con herramientas gráficas para la gestión y control de versiones con git y github.



- **Open Source:** Es una herramienta de código abierto.
- **Comunidad:** La comunidad de usuarios es muy grande por lo que el soporte está garantizado.
- **Themes:** Variedad de temas tanto claros como oscuros.
- **Soporte a diferentes lenguajes:** PHP, SQL, Java, JavaScript, TypeScript, Go, C#, Visual Basic, F#, Lua, Perl, C++, Objective C, Ruby, Lees, Sass, Swift, Python, R, etc.
- **Extensiones:** variedad de plugins desarrollados por la comunidad como por las empresas de software oficiales.
  - **Auto Close Tag:** Desde VScode 1.16, tiene soporte integrado de etiqueta cerrada para archivos HTML, Handlebars y Razor. Esta extensión está habilitada para otros lenguajes como XML, PHP, Vue, JavaScript, TypeScript, JSX, TSX, etc.
  - **Auto Rename Tag:** Un plugin bastante sencillo, pero al mismo tiempo muy útil, cambia automáticamente el nombre de una etiqueta HTML/XML emparejada. Eso es todo.
  - **Bookmarks:** Ayuda a programador a navegar en su código, moviéndose entre porciones importantes del mismo de manera fácil y rápida. Entre algunas características se pueden mencionar:
    - Marcado y desmarcado de código.
    - Posibilidad de dar un nombre identificativo a porciones de código.
    - Saltos entre marcadores.
    - Ver los marcadores organizados en un solo archivo.
    - Barra lateral dedicada, etc.
  - **GitLens:** Ayuda a comprender mejor el código . Echa un vistazo rápidamente a quién, por qué y cuándo se cambió una línea o un bloque de código. Retroceda a través de la historia para obtener más información sobre cómo y por qué evolucionó el código. Explore sin esfuerzo la historia y la evolución de un código base
  - **IntelliSense for CSS class names in HTML:** Auto completado de nombres de clases que se encuentran en un espacio de trabajo previamente definido. Entre algunas peculiaridades del plugin se pueden resaltar:
    - Funciona con hojas de estilos externas anexadas con el atributo link
    - Es posible almacenar las clases en cache para un mejor funcionamiento
  - **Prettier - Code Formatter:** Formatea el código

## 4 LENGUAJE JAVASCRIPT:

JavaScript es un potente lenguaje de programación basado en scripts y enfocado a Internet, que se utiliza fundamentalmente para:



- realización de aplicaciones para la www.
- creación de interfaces de usuario.
- añadir interactividad a las páginas web.
- generación de HTML dinámico.
- control de eventos en páginas HTML
- etcétera.

A menudo se llama a JavaScript "lenguaje de escritura de guiones". Un guión JavaScript es un programa incluido en una página HTML, el texto del guión no aparece en la pantalla del usuario. Los programas se sitúan en una página web entre las etiquetas `<SCRIPT>` y `</SCRIPT>`, y se ejecutarán automáticamente al abrir dicha página con un navegador. No se requiere compilador. Será el propio navegador el encargado de interpretar y ejecutar el código JavaScript.

#### Características del lenguaje JavaScript:

- Es un lenguaje sencillo.
- Diseñado para desarrollar aplicaciones para la www.
- Lenguaje interpretado por el navegador. No requiere compilador.
- No requiere declarar variables ni tipos (aunque es aconsejable hacerlo).

## 5 JAVASCRIPT Y HTML:

Los programas JavaScript se encuentran en páginas HTML. El navegador reconoce un programa JavaScript cuando se encuentra con una etiqueta `<SCRIPT>`. A partir de ese momento será el intérprete JavaScript el encargado de interpretar y ejecutar el código hasta la etiqueta `</SCRIPT>`.

### 5.1 PASOS PARA CREAR UN PROGRAMA JAVASCRIPT:

1. Crearemos una página HTML utilizando cualquier editor, por ejemplo el *Bloc de Notas*. (También podemos utilizar una página ya creada y continuar con el paso siguiente).
2. Insertaremos dentro de la página HTML las etiquetas `<SCRIPT>` y `</SCRIPT>` en el lugar donde se situará el código JavaScript.
3. Introduciremos el código JavaScript entre dichas etiquetas.
4. Salvaremos el programa poniendo especial cuidado para que el editor mantenga la extensión **.htm**
5. Para ejecutar el programa solamente tendremos que abrir el archivo con el navegador.

### 5.2 DÓNDE COLOCAR LOS SCRIPTS:

Los scripts se pueden colocar en dos sitios de la página HTML:

- En la cabecera de la página entre las etiquetas `<HEAD>` y `</HEAD>` o
- En el cuerpo de la página entre las etiquetas `<BODY>` y `</BODY>`.



Cuando el script se incluye en la cabecera de una página éste suele estar estructurado en forma de funciones que se ejecutan sólo cuando se las invoca desde otro script o cuando se produce el evento al que están asociadas.

Cuando el script se incluye en el cuerpo de una página, las instrucciones que lo forman se ejecutan en el momento de la carga de la página en el navegador.

***Ejemplo El siguiente script incluye una función en la cabecera del documento HTML, desde el script del cuerpo de la página se llama a la función en el momento de la carga de la página en el navegador:***

```
<HTML>
<HEAD><TITLE>Mi primer guión</TITLE>

<SCRIPT LANGUAGE="JavaScript">
function vertexto(){
    document.write("Esto es JavaScript.")
}
</SCRIPT>

</HEAD>
<BODY>
Esto es HTML<br>
<b>
<SCRIPT LANGUAGE="JavaScript">
    vertexto()
</SCRIPT>
</b>
<br>Esto es HTML
</BODY>
</HTML>
```

### 5.3 Comentarios en los guiones:

JavaScript permite añadir comentarios al código que no serán ejecutados ni visualizados por el navegador. Estos comentarios sirven para explicar el código del programa; por ejemplo, para aclarar una expresión compleja, para indicar cuál es el contenido de una variable, etcétera. Se visualizarán cuando editemos el archivo HTML.

Los comentarios pueden ser:



- **De una línea.** Se indican mediante dos barras inclinadas a la derecha (/). Todo lo que aparezca detrás de dichas barras hasta el final de la línea será considerado comentario y, por tanto, será ignorado por el navegador.
- **De varias líneas.** En este caso deberemos indicar el comienzo del comentario mediante los caracteres barra inclinada a la derecha y asterisco (/\*). También indicaremos el final del comentario, en este caso mediante la secuencia de caracteres inversa: asterisco y barra inclinada a la derecha (\*/).