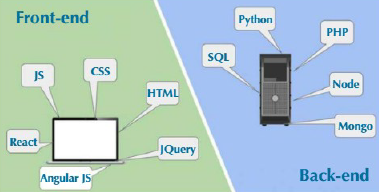
* 1. **Introduccion**

**Cuando se creó la web allá en 1989 por Tim Berners-Lee, nadie se podría imaginar en lo que se convertiría.**

**El W3C es el consorcio que se encarga de desarrollar los estándares para que el desarrollo de internet funcione. En España, es** [**www.w3c.es**](http://www.w3c.es)**.**

1. **Front-end y back-end**

**back-end (parte no visible de la web, como bases de datos o scripts que se ejecutan en el servidor) y front-end (parte visible de una web, como hojas de estilo, código HTML, scripts que se ejecutan en el cliente).**

****

**Los técnicos de back-end se encargan de todo el proceso del lado del servidor, como el acceso a la base de datos (MySQL, MaríaDB, etc), creación de servicios, etc. Estos técnicos programarán en lenguajes como PHP, Ruby on Rails, Node.js, etc.**

**Los frameworks nacieron como librerías, que tenían unas estructuras que permitían al programador tener una base para la creación y el desarrollo de sus proyectos. Hoy pueden utilizar lenguajes como TypeScript o JSX, los cuales se compilan a JavaScript.**

**Ventajas:**

* **El coste. Muchos de estos frameworks son de código abierto.**
* **Están probados y suele carecer de errores, puesto que muchos programadores lo utilizan. Además, alto nivel de seguridad y rendimiento.**
* **Permite desarrollar más rápido, muchas de las estructuras, clases, etc., vienen ya incorporadas en el framework.**
* **Cualquier persona que maneja un framework puede entender e incorporarse de forma eficiente a un equipo que lo esté utilizando.**

**el front-end, es la parte creativa y original, puesto que es más cercano al diseñador, aunque también se trabaja en el código. La programación de la interfaz se llevará en HTML, CSS, JavaScript. Con frameworks como AngularJS o ReactJS, el front-end está ganando terreno a la parte servidora. En muchas ocasiones, estos frameworks sirven para almacenar estructuras de datos relegando al back-end.**

1. **Lenguajes de programación en entorno cliente**

**se basa en tres pilares fundamentales:**

**1. El lenguaje HTML. Es un lenguaje de marcado. Define el contenido que va a tener el documento. La función del navegador web será la de leer e interpretar todo este contenido, junto con las etiquetas, y visualizarlo al cliente. Cualquier navegador debería visualizar el contenido de la página web con el mismo aspecto.**

**2. El lenguaje CSS. Lenguaje de diseño gráfico y su objetivo es que la página web sea atractiva al usuario. No modifica el comportamiento ni el contenido, sino el aspecto de la página web.**

**3. El lenguaje JavaScript. Agrega contenido dinámico a las páginas web. Verdadero lenguaje de programación, a diferencia HTML y CSS.**

**Actualmente, las empresas programan basándose en frameworks de JavaScript.**

* + 1. **ReacJS**

**Framework creado por Facebook que permite a los programadores realizar aplicaciones web de una forma rápida y eficiente renderizando (dibujando) los componentes del front-end de forma sencilla y eficaz.**

**Programación orientada a componentes (no objetos). Los componentes gestionan su propio estado y, cuando se agrupa una serie de componentes, los programadores son capaces de ir creando las interfaces de usuario.**

**Utiliza un DOM virtual que mapea los objetos desde este hasta el DOM del navegador.**

* + 1. **AngularJS**

**Creado y es mantenido por Google. El problema con este framework es que no son fáciles de aprender.**

**Actualmente, se programa en TypeScript, que es un superconjunto de JavaScript desarrollado por Microsoft y utiliza el patrón reactivo RxJS.**

* + 1. **Vue.js**

**Una de las características es la ligereza y velocidad de ejecución. El objetivo fue el crear un framework con las mejores ventajas de los existentes. A diferencia de Angular, es más fácil de aprender.**

**Al igual que React, utiliza un DOM virtual, dadas las ventajas que ofrece este tipo de implementaciones.**

1. **Características de los lenguajes de script**

**Hubo un tiempo en el que los lenguajes de programación se utilizaban para crear programas de nóminas, contabilidad, edición de texto, etc.**

**Ahora se necesitan aplicaciones que se puedan ejecutar sobre un navegador web o aplicaciones para dispositivos (smartphones o tabletas).**

**Ya no se desarrolla todo desde cero, sino que, se utiliza un sistema host – navegador - para integrar pequeños fragmentos de código que aporten el aspecto dinámico a las páginas web.**

**Los scripts nacieron como fragmentos de código que realizaban ciertas tareas. En los sistemas operativos, los scripts se usan para automatizar tareas y son ejecutadas por un intérprete de comandos (los scripts son interpretados).**

**Actualmente, no son pequeños fragmentos de código, sino que pueden considerarse programas.**

**diferencias entre los lenguajes de script y de programación:**

* **Los de script son interpretados, mientras que, los de programación se compilan.**
* **Los de script usan componentes ya preexistentes, los de programación, empiezan a desarrollarse desde cero.**
* **los scripts se incrustan dentro de otros programas (JavaScript se integra con el HTML).**
* **Los programas de lenguajes de programación se pueden ejecutar de forma independiente, un lenguaje de script se ejecuta dentro de otro programa.**
* **Los scripts se ejecutan línea a línea, pueden producirse errores en ejecución.**
* **Los de script no generan un fichero ejecutable.**
* **Los de script no necesitan ser compilados.**
* **Los de script han sido diseñados para que ser fáciles de utilizar y programar.**

**lenguajes de programación: Java, C, C++, Swift, etc.**

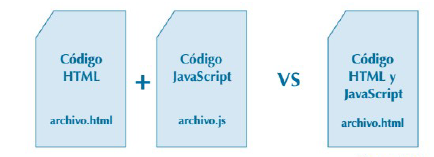
**lenguajes de scripting: JavaScript, Shell, Perl, PHP, Python, Ruby, etc.**

1. **Integración de JavaScript de html**

**JavaScript se combina al código HTML. Dos opciones:**

**1. Integrar JavaScript dentro de los archivos HTML.**

**2. Tener separado del HTML el código JavaScript en archivos con extensión js.**

****

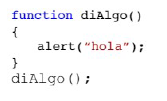
* + 1. **Ejemplo de JavaScript en ficheros js separados**

**Los archivos JavaScript en un proyecto profesional suelen estar en ficheros separados del HTML. Ej:**

****

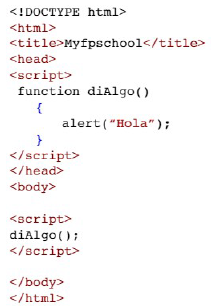
**La etiqueta <script src="script.js"></script> indica que el código JavaScript está contenido en un archivo aparte llamado script.js.**

**Contenido del archivo script.js:**

****

* + 1. **Ejemplo de JavaScript con código del html**

**JavaScript también puede estar embebido dentro de HTML. Ej;**

****

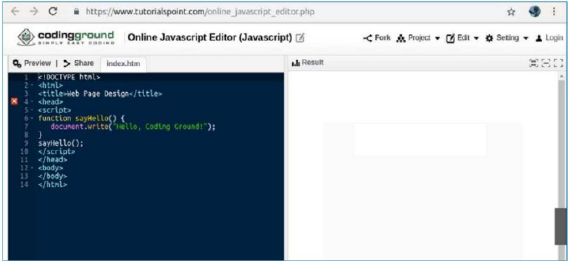
**Las ventajas de tener el JavaScript en un fichero separado: páginas cargarán mucho más rápido, se independiza el HTML del código y, el JavaScript será más fácil de mantener.**

1. **Herramientas de programación en javascript**

**Puede servir un simple editor de texto. Esto no es lo deseable cuando se desarrolla un proyecto, las empresas suelen utilizar otras alternativas.**

1. **Herramientas online**

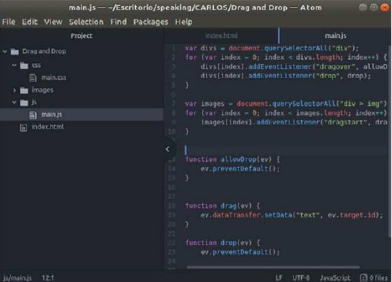
**Las ventajas que ofrece un IDE (entorno integrado de desarrollo) online: se puede ejecutar y probar código desde cualquier dispositivo teniendo acceso a internet. Ej: Coding Ground de Tutorialspoint**

****

**Esta herramienta no solo permite gestionar los distintos ficheros, sino que se puede descargar o agregar cualquier fichero que haga falta.**

1. **Utilización de IDE y sistemas de control de versiones**

**Una alternativa puede ser utilizar Atom. Está disponible para Linux, Windows y Mac OS X y es una herramienta libre, gratuita, ligera y potente.**

****

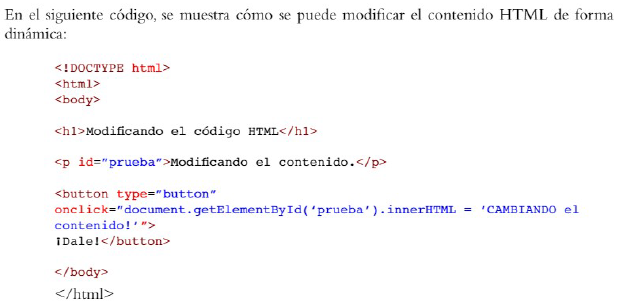
**Algunas ventajas:**

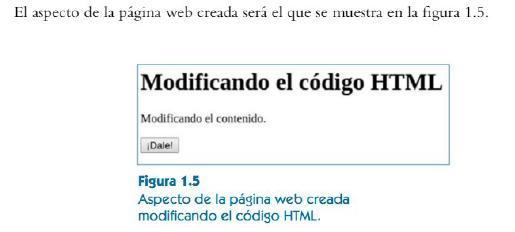
* **Es open-source, si se desea hacer modificaciones en el código fuente, está disponible para todo el público.**
* **Es gratuito.**
* **Tiene una gran comunidad detrás de la herramienta,**
* **Es modular. Se pueden deshabilitar funciones y reemplazar por otras.**
* **Tiene integrado un package manager. Se pueden instalar paquetes.**
* **Está integrado con GitHub.**
* **El sistema de paneles es muy apreciado por los programadores.**
* **Sistema muy bueno de autocompletado.**

1. **Posibilidades que ofrece javascript**

**se mostrarán varios ejemplos de qué se puede hacer con JavaScript.**

1. **Modificación del contenido de una pagina web**

****

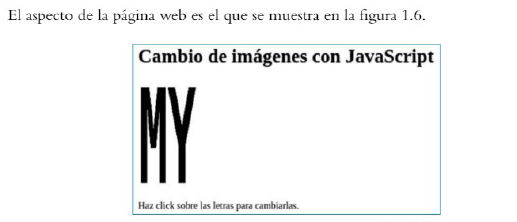
****

**ejecutando el código anterior, al pulsar el botón ¡Dale!, cambiará el párrafo "Modificando el contenido" por el párrafo "Cambiando el contenido".**

1. **Cambiar atributos de objetos html**

**se va a modificar de forma dinámica el atributo src de un objeto tipo img en un código HTML.**

****

****

**cuando se hace clic sobre las letras MY, estas cambian de negro a verde, y viceversa, porque, en el script anterior, comprueba qué imagen se está visualizando.**

1. **Cambiar el estilo CSS**

**Con JavaScript no solamente se puede cambiar el contenido, también se puede cambiar el estilo (CSS) de cualquier elemento.**

**se va a generar una página cuyo aspecto sea el de la figura 1.7 y, una vez que se pulse un botón, cambie el aspecto del párrafo para parecerse a la figura 1.8.**

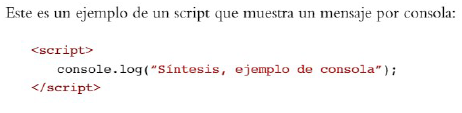
****

****

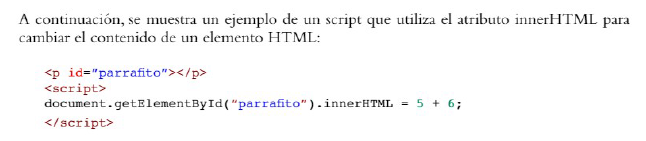
**en la función myFunction, se selecciona el párrafo que se quiere modificar y se cambia su color a rojo y su tamaño a 25 píxeles.**

* 1. **Comunicación de javascript con el exterior**

**Existen varias opciones para que el código JavaScript se comunique con el usuario. Ej:**

****

1. **Escribir en cualquier elemento html utilizando el atributo innerHTML**

****

**se ha seleccionado previamente el objeto con identificador parrafito utilizando el método getElementById**

1. **Generar directamente HTML utilizando el método document.write()**

**Con el método write se puede generar código html directamente. Se pueden añadir etiquetas en la llamada al método.**

****

1. **Generar un mensaje de alerta utilizando el método window.alert()**

**Con el método window.alert() o alert(), se puede mostrar un diálogo emergente en el navegador.**

**ejemplo del método window.alert() dentro de un código JavaScript:**

****

****

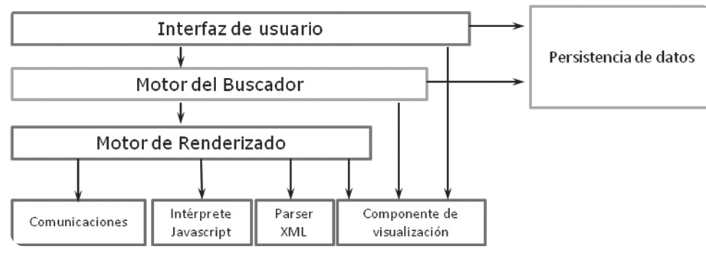
**Estructura de un navegador.**

**Las principales partes de un navegador web moderno son:**

**La Interfaz de usuario**

**motor de renderizado (el verdadero núcleo del navegador)**

**intérprete JavaScript**

****

**Interfaz Gráfica de Usuario**

**Es el medio por el cual interactuamos con la aplicación. Su función es desarrollar la comunicación con la aplicación de la forma más fácil y cómoda.**

**Motor de renderizado**

**Es el componente más importante de cualquier navegador. 'Dibuja' el contenido en una ventana, que es mostrada al usuario.**

**Webkit. es un motor de renderizado de código abierto para navegadores web, desarrollado por Apple. Incluye dos frameworks de más bajo nivel:**

**WebCore: analizador sintáctico y motor de renderizado HTML**

**JavaScriptCore: intérprete de JavaScript basado en KJS**

**motores JavaScript**

**Parte del navegador que interpreta y ejecuta el código escrito en el lenguaje de programación JavaScript. Son exclusivos de cada navegador.**