

1.1. Introducción

Cuando se creó la web allá por 1989 en el laboratorio europeo de partículas (CERN) por Tim Berners-Lee, nadie se podría imaginar en lo que se convertiría décadas más tarde.

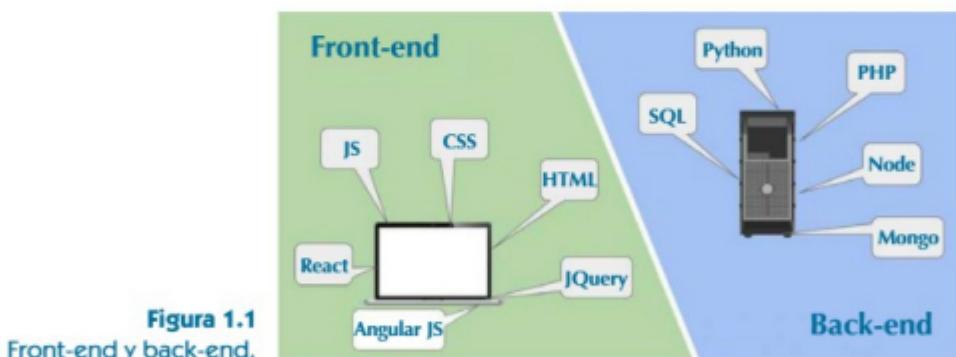
Actualmente, el W3C es el consorcio que se encarga de desarrollar los estándares para que el desarrollo de internet funcione a largo plazo. El W3C es un organismo abierto y cualquiera puede unirse a sus grupos y participar en los blogs u otras discusiones. En España, el sitio web del W3C es www.w3c.es.

Hoy en día, gran parte del comercio mundial se realiza de forma electrónica y no solamente Amazon, Alibaba u otros gigantes, sino cualquier pequeño y mediano comercio. Es más, dada la deslocalización de sus clientes o trabajadores, muchas empresas están migrando sus sistemas tradicionales dentro de los servidores de la empresa a aplicaciones alojadas en servidores de internet (computación en la nube). Por ello, empresas como AWS (Amazon Web Services) están creciendo de forma exponencial. Por estas y otras razones, el puesto de desarrollador web está siendo muy demandado en el mercado.

El desarrollo web es muy diverso y, por ello, los trabajadores se suelen especializar solo en un aspecto de este. Existen diseñadores web en los que se valora la originalidad o el buen gusto para elegir colores, formas y disposición de elementos en una página web, programadores web expertos en algún lenguaje del lado del cliente o servidor, administradores de sistemas y bases de datos, arquitectos web, etc.

1.2. Front-end y back-end

Actualmente, se suele catalogar el desarrollo web en dos partes back-end (la parte no visible de la web, como las bases de datos o los scripts que se ejecutan en el servidor) y front-end (la parte visible de una web, como las hojas de estilo, el código HTML, los scripts que se ejecutan en el lado del cliente). A continuación, se describirá con más profundidad este modelo cliente servidor.



En las empresas desarrolladoras de aplicaciones web, suele haber técnicos especialistas en back-end y front-end. Los técnicos de back-end se encargan de todo el proceso en el lado del servidor, como el acceso a la base de datos (MySQL, MaríaDB, PostgreSQL, MongoDB, Oracle), creación de servicios, etc. Estos técnicos programarán en lenguajes como PHP, Ruby on Rails, Django, Node.js, .NET, etc.



PARA SABER MÁS

Los frameworks nacieron como librerías, más o menos completas, que tenían una serie de estructuras que permitían al programador tener una base para la creación y el desarrollo de sus proyectos. Hoy, son mucho más que eso, puesto que pueden utilizar lenguajes como TypeScript o JSX, los cuales luego se compilan a JavaScript.

Entre las ventajas que aportan los frameworks, están:

- **El coste.** Muchos de estos frameworks son de código abierto, con lo cual no hay que realizar ninguna inversión.
- Están suficientemente *probados* y su código suele carecer de errores, puesto que muchos programadores lo utilizan. Además, suelen tener un alto nivel de seguridad y rendimiento.
- Permite desarrollar mucho más *rápido*, puesto que muchas de las estructuras, clases, patrones de diseño, etc., vienen ya incorporadas en el framework.
- Cualquier persona que maneja un framework determinado *puede entender* e incorporarse de forma eficiente y efectiva a un equipo de desarrollo que lo esté utilizando en un proyecto determinado.

En la parte del front-end, prima la parte creativa y la originalidad, puesto que el perfil es más cercano al diseñador, aunque también se trabaja en el código. La programación de la interfaz se llevará a cabo en lenguajes como HTML, CSS, JavaScript. Con frameworks como AngularJS, ReactJS y otros, la parte cliente o front-end está ganando terreno a la parte servidora. En muchas ocasiones, estos frameworks están diseñados para almacenar estructuras de datos relegando al back-end o simplemente para asegurar la persistencia de estos.

1.3. Lenguajes de programación en entorno cliente

La programación web en el lado del cliente se basa en tres pilares fundamentales que se citan a continuación:

1. *El lenguaje HTML.* No es un lenguaje de programación, sino un lenguaje de marcado. El HTML 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. La ventaja del código HTML es que cualquier navegador debería visualizar el contenido de la página web de la misma forma y con el mismo aspecto.
2. *El lenguaje CSS.* Define la presentación del documento. CSS es un 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.* El código o lenguaje JavaScript agrega el contenido dinámico a las páginas web. JavaScript es un verdadero lenguaje de programación, a diferencia de los lenguajes HTML y CSS.

Actualmente, las empresas no programan directamente sobre JavaScript, sino basándose en un framework de este. A continuación, se citarán algunos de los frameworks más utilizados para JavaScript.

1.3.1. ReactJS

ReactJS (<https://reactjs.org/>) es un 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 una manera sencilla y eficaz.

React utiliza programación orientada a componentes (que 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.

Una de las características de React es que utiliza un DOM virtual que mapea los objetos desde este hasta el DOM del navegador.



Actividad propuesta 1.1

Averigua qué es la programación reactiva.

1.3.2. AngularJS

Angular (<https://angular.io>) fue creado y es mantenido por Google. La primera versión de Angular se denominó *AngularJS* y todavía hay una comunidad utilizando este framework por su fácil integración con JavaScript. Las versiones sucesivas de Angular se denominan *Angular* a secas y ya ha dejado de ser una simple librería para pasar a ser una plataforma de desarrollo.

El problema con este y otros frameworks es que su curva de aprendizaje es bastante pronunciada, dado que no son fáciles de aprender.

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

1.3.3. Vue.js

Una de las características de este framework (<https://vuejs.org>) frente a otros es la ligereza y la velocidad de ejecución. El objetivo que se plantearon sus desarrolladores y diseñadores fue el crear un framework con las mejores ventajas de los existentes. A diferencia de Angular, su curva de aprendizaje no es tan pronunciada y los desarrolladores de Laravel (un framework de back-end) lo utilizan para usarlo en el front-end de sus aplicaciones.

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

**TOMA NOTA**

Existen otros frameworks como EmberJS, BackboneJS, MeteorJS, Aurelia.js, Polymer o Mithril.js. Escoger un framework es un proceso importante, puesto que el objetivo es elegir el más popular para que tenga un soporte más amplio, el que tenga más futuro, el más rápido de ejecución, el más fácil de aprender. Dicha elección muchas veces es complicada, ya que la tecnología cambia de forma muy rápida.

www**Recurso web**

A través del siguiente código QR, puedes acceder a la página web Raygun y encontrar información sobre los nueve frameworks más utilizados:



1.4. 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, hojas de cálculo, gestión de almacenes, etc., en los que se realizaba un análisis, un diseño de la aplicación y luego se pasaba a codificar todo lo desarrollado en fases anteriores. Los programas se ejecutaban en un equipo standalone o en entornos cliente servidor.

Desde hace mucho tiempo, ese tipo de necesidades han cambiado, puesto que la conectividad total y las nuevas necesidades de las empresas y clientes son distintas. Ahora se necesitan aplicaciones que se puedan ejecutar sobre un navegador web o aplicaciones para dispositivos móviles como smartphones o tabletas.

Ya no se desarrolla todo desde cero, sino que, en muchos casos, se utiliza un sistema host —como puede ser el navegador— para integrar pequeños fragmentos de código que aporten el aspecto dinámico a las páginas web visualizadas.

Los scripts nacieron como fragmentos de código que realizaban ciertas tareas o rutinas concretas. En los sistemas operativos, los scripts se usan para automatizar tareas y siempre van a ser ejecutadas por un intérprete de comandos (por lo tanto, los scripts siempre van a ser interpretados).

Actualmente, los scripts no son pequeños fragmentos de código, sino que pueden considerarse auténticos programas.

Véanse algunas de las diferencias entre los lenguajes de script y los lenguajes de programación:

- Los lenguajes de script son interpretados, mientras que, muchas veces, los lenguajes de programación se compilan.
- Los lenguajes de script utilizan componentes ya preexistentes, mientras que los lenguajes de programación, en ocasiones, empiezan a desarrollarse desde cero.
- A veces, los scripts se incrustan dentro de otros programas (como el JavaScript se integra con el HTML).

- Los programas de lenguajes de programación se pueden ejecutar de forma independiente, mientras que un lenguaje de script se ejecuta muchas veces dentro de otro programa.
- Los scripts se ejecutan línea a línea, con lo cual pueden producirse muchos errores en ejecución.
- Los lenguajes de script no generan un fichero ejecutable.
- Los lenguajes de script no necesitan ser compilados.
- Los lenguajes de script han sido diseñados para que ser fáciles de utilizar y programar.

Algunos lenguajes de programación son: Java, C, C++, Swift, Pascal, etc.

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



SABÍAS QUE...

El lenguaje de scripting Python es uno de los que tienen más proyección porque es ampliamente utilizado en inteligencia artificial.

1.5. Integración de JavaScript dentro de HTML

Como ya se ha explicado, JavaScript se combina o complementa al código HTML de una página web. Existen dos opciones:

1. Integrar el código JavaScript dentro de los archivos HTML.
2. Tener separado del HTML el código JavaScript en archivos con extensión js.



Figura 1.2

Código JavaScript dentro y fuera de un archivo HTML.

TOMA NOTA



Lo más limpio y eficaz es tener el código JavaScript fuera de los archivos HTML por las ventajas que ofrece.

A continuación, en el apartado 1.5.1, se desarrolla un ejemplo de ambas posibilidades.

1.5.1. Ejemplo de JavaScript en ficheros js separados

Los archivos JavaScript en un proyecto profesional suelen estar en ficheros separados del código HTML. Generalmente, las empresas y programadores con experiencia así lo aconsejan. Seguidamente, se ofrece al lector un ejemplo de página web en el que los códigos HTML y JavaScript residen en ficheros diferentes.

RECUERDA

- ✓ El código JavaScript está contenido dentro de las etiquetas `<script>` y `</script>`.

El código JavaScript se puede colocar tanto dentro de la etiqueta `<head>` como dentro de la etiqueta `<body>`. Nuestro consejo es que se sitúe en el mismo sitio. No es buena práctica diseminar el código por toda una página porque luego es imposible de entender y mantener.

Es posible ver versiones antiguas de JavaScript con etiquetas del tipo: `<script type="text/javascript">`.

Contenido del archivo `index.html`:

```
<!DOCTYPE html>
<html>
<title>Myfpschool</title>
<head>
<script src="script.js"></script>
</head>
<body>
</body>
</html>
```

La etiqueta `<script src="script.js"></script>` indica que el código JavaScript está contenido en un archivo aparte llamado `script.js`, que se encuentra en la misma carpeta que este archivo `index.html`.

A continuación, se muestra el contenido del archivo `script.js`:

```
function diAlgo()
{
    alert("hola");
}
diAlgo();
```

Actividad propuesta 1.2



Crea los dos archivos de este apartado y ejecútalos comprobando que la página web muestra un mensaje al cargarse.

1.5.2. Ejemplo de JavaScript con código dentro del HTML

Como se ha explicado anteriormente, el código JavaScript también puede estar embebido dentro del código HTML. A continuación, se muestra el mismo ejemplo del apartado 1.5.1, pero con los códigos HTML y JavaScript en un mismo fichero.

Detalle del archivo *index.html*:

```
<!DOCTYPE html>
<html>
<title>Myfpschool</title>
<head>
<script>
    function diAlgo()
    {
        alert("Hola");
    }
</script>
</head>
<body>

<script>
diAlgo();
</script>

</body>
</html>
```



Actividad propuesta 1.3

Si creas y ejecutas las actividades propuestas 1.1 y 1.2, podrás comprobar que el funcionamiento es igual en ambos. Observa cómo ambos códigos muestran un mensaje de alerta al navegarante.

Como se puede observar, aunque el resultado es el mismo, la forma de distribuir el código HTML y el JavaScript no es igual. El mejor consejo es utilizar archivos separados para HTML y JavaScript, salvo que las líneas de código de este último sean mínimas y no vayan a ser modificadas prácticamente nunca.

Las ventajas de tener el JavaScript en un fichero separado son que las páginas cargarán mucho más rápido, se independiza el HTML del código y, como se puede adivinar, el JavaScript será mucho más fácil de mantener.

TOMA NOTA



Los proyectos suelen tener los scripts en una carpeta aparte de nombre **script** o **js** para tener los archivos más ordenados.

1.6. Herramientas de programación en JavaScript

Existen múltiples alternativas a la hora de elegir una herramienta de programación. Una de las ventajas de programar en JavaScript es que puede servir un simple editor de texto como Notepad o gedit. Esto no es lo deseable cuando se desarrolla un proyecto y, por lo tanto, las empresas y profesionales suelen utilizar otras alternativas como los entornos de programación online o herramientas de programación con sistemas de control de versiones.

A continuación, se mostrarán dos alternativas de herramientas de programación.

1.6.1. Herramientas online

Las ventajas que ofrece un IDE (entorno integrado de desarrollo) online son evidentes, se puede ejecutar y probar código desde cualquier dispositivo solamente teniendo acceso a internet. Existen muchas herramientas en el mercado y una de ellas es Coding Ground de Tutorialspoint (https://www.tutorialspoint.com/online_javascript_editor.php).

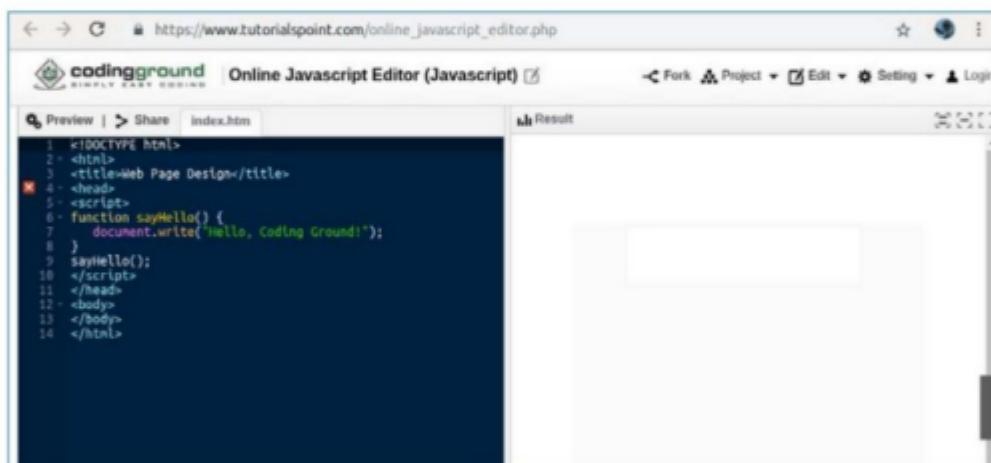


Figura 1.3
Interfaz de Coding Ground, herramienta para compilar JavaScript online.

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

1.6.2. Utilización de IDE y sistemas de control de versiones

Otra de las opciones es utilizar un IDE complementado con un sistema de control de versiones. Una alternativa puede ser utilizar Atom (<https://atom.io>).

Atom está disponible para plataformas Linux, Windows y Mac OS X y es una herramienta libre, gratuita, ligera y potente.

The screenshot shows the Atom IDE interface. At the top, there's a menu bar with File, Edit, View, Selection, Find, Packages, and Help. Below the menu is a toolbar with icons for Undo, Redo, Save, and others. The main area is divided into two panes: the left pane shows a project structure for a 'Drag and Drop' application, containing subfolders like 'css', 'images', and 'js', and files like 'main.css', 'index.html', and 'main.js'. The right pane is a code editor with the file 'main.js' open. The code in 'main.js' is as follows:

```

main.js — ~/Escritorio/speaking/CARLOS/Drag and Drop — Atom
File Edit View Selection Find Packages Help
Project index.html main.js
Drag and Drop
  css
    main.css
  images
  js
    main.js
  index.html
var divs = document.querySelectorAll("div");
for (var index = 0; index < divs.length; index++) {
  divs[index].addEventListener("dragover", allowDrop);
  divs[index].addEventListener("drop", drop);
}
var images = document.querySelectorAll("div > img");
for (var index = 0; index < images.length; index++)
  images[index].addEventListener("dragstart", drag);
function allowDrop(ev) {
  ev.preventDefault();
}
function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}
function drop(ev) {
  ev.preventDefault();
}

```

At the bottom of the code editor, it says 'js/main.js 121' and shows encoding as 'UTF-8' and language as 'JavaScript'. There are also tabs for 'File' and 'Edit'.

Figura 1.4
Detalle de un IDE
utilizando Atom.

Se citan algunas ventajas de este IDE:

- Es open-source, lo que quiere decir que, si se desea hacer modificaciones o arreglar el código fuente, está disponible para todo el público. Muchas veces los programadores no van a modificar el código, pero al ser software libre se aseguran de que una comunidad de usuarios se encargará de ello, haciendo que los errores y mejoras se corrijan mucho antes que con un software propietario.
- Es gratuito.
- Tiene una gran comunidad detrás de la herramienta, con lo cual cualquier problema o inquietud puede ser resuelta en Slack o incluso en Twitter.
- Es modular. Se pueden deshabilitar ciertas funciones y reemplazar por otras del gusto del programador.
- Tiene integrado un package manager. Se pueden instalar paquetes y temas con un comando `apm` desde el terminal.
- Está integrado con GitHub. Atom está pensado para utilizarlo conjuntamente con este sistema tan famoso de control de versiones.
- El sistema de paneles de Atom es muy apreciado por los programadores.
- Tiene un sistema muy bueno de autocompletado.

Recursos web



Como alternativa a Atom, se puede usar Sublime text. (<https://www.sublimetext.com/>) o VisualStudio (<https://code.visualstudio.com>).

1.7. Posibilidades que ofrece JavaScript

A continuación, se mostrarán varios ejemplos básicos de qué se puede hacer con JavaScript.

1.7.1. Modificación del contenido de una página web

En el siguiente código, se muestra cómo se puede modificar el contenido HTML de forma dinámica:

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

<h1>Modificando el código HTML</h1>

<p id="prueba">Modificando el contenido.</p>

<button type="button"
onclick="document.getElementById('prueba').innerHTML = 'CAMBIANDO el
contenido!' ">
¡Dale!</button>

</body>
</html>
```

El aspecto de la página web creada será el que se muestra en la figura 1.5.

Modificando el código HTML

Modificando el contenido.

Figura 1.5
Aspecto de la página web creada
modificando el código HTML.

Como se puede observar, ejecutando el código anterior, al pulsar el botón ¡Dale!, cambiará el párrafo “Modificando el contenido” por el párrafo “Cambiando el contenido”. Más adelante se estudiará en profundidad el código JavaScript del ejemplo anterior.

1.7.2. Cambiar atributos de objetos HTML

En el siguiente ejemplo, se va a modificar de forma dinámica el atributo src de un objeto tipo img en un código HTML.

Tras copiar el siguiente código en un archivo de texto, puede observarse cómo funciona:

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

<h1>Cambio de imágenes con JavaScript</h1>



<p>Haz click sobre las letras para cambiarlas.</p>

<script>
function cambiaPic() {
    var image = document.getElementById('myFPIImage');
    if (image.src.match("green")) {
        image.src = "http://myfpschool.com/wp-content/uploads/2016/06/my-
        black.jpeg";
    } else {
        image.src = "http://myfpschool.com/wp-content/uploads/2016/06/my-
        green.jpeg";
    }
}
</script>

</body>
</html>
```

El aspecto de la página web es el que se muestra en la figura 1.6.



Figura 1.6
Aspecto de la página web del cambio
de imágenes con JavaScript.

Se puede observar que, 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.



Actividad propuesta 1.4

Copia el código del ejemplo de este apartado y comprueba si funciona en un navegador. En esta ocasión, utiliza tus propias imágenes.

1.7.3. Cambiar el estilo CSS

Con JavaScript no solamente se puede cambiar el contenido, sino que también se puede cambiar el estilo (CSS) de cualquier elemento de la página HTML.

Para ver cómo se puede cambiar el aspecto, 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.



Figura 1.7
Aspecto de la página antes
de pulsar el botón.



Figura 1.8
Aspecto de la página
después de pulsar el botón.

Para ver cómo ha cambiado el aspecto de la página anterior, hay que copiar el siguiente código en un editor como Atom y comprobar si funciona correctamente:

```
<!DOCTYPE html>
<html>
<body>
<p id="mytxt">Aprende JavaScript con MyFPSchool!</p>
<button type="button" onclick="myFunction()">¡Dale!</button>
<script>
function myFunction() {
    var x = document.getElementById("mytxt");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>
</body>
</html>
```

Como se puede observar, 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.

1.8. Comunicación de JavaScript con el exterior

Existen varias opciones para que el código JavaScript se comunique con el usuario. A continuación, se detalla cada una de ellas.

1.8.1. Escribir en la consola del navegador utilizando console.log()

Esta opción la utilizan solamente los desarrolladores. Ningún usuario suele acceder a la consola para ver el contenido escrito en ella.

RECUERDA

- ✓ Para acceder a la consola, basta con pulsar F12 y elegir Consola en su navegador.

Este es un ejemplo de un script que muestra un mensaje por consola:

```
<script>
    console.log("Síntesis, ejemplo de consola");
</script>
```

1.8.2. Escribir en cualquier elemento HTML utilizando el atributo innerHTML

A continuación, se muestra un ejemplo de un script que utiliza el atributo innerHTML para cambiar el contenido de un elemento HTML:

```
<p id="parrafito"></p>
<script>
document.getElementById("parrafito").innerHTML = 5 + 6;
</script>
```

Nótese que se ha seleccionado previamente el objeto con identificador parrafito utilizando el método `getElementById`.

1.8.3. Generar directamente HTML utilizando el método document.write()

Con el método write se puede generar código HTML directamente. Se pueden también añadir etiquetas en la llamada al método si se desea.

```
<script>
    document.write("<h2>Buenos días</h2>");
</script>
```

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

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

A continuación se presenta un ejemplo de utilización del método `window.alert()` dentro de un código JavaScript:

```
<script>
    window.alert("BUENAS NOCHES");
</script>
```



Figura 1.9

Ejemplo del diálogo emergente obtenido usando el método `window.alert()` dentro de un código JavaScript.

Resumen

- W3C es el que se encarga de desarrollar los estándares para que el desarrollo de internet funcione a largo plazo.
- Los frameworks nacieron como librerías, más o menos completas, que tenían una serie de estructuras que permitían al programador disponer de una base para la creación y el desarrollo de sus proyectos.
- Los técnicos de back-end se encargan de todo el proceso en el lado del servidor, como el acceso a la base de datos.
- En la parte del front-end, prima la parte creativa y la originalidad, puesto que el perfil es más cercano al diseñador, aunque también se trabaja en el código.
- ReactJS es un framework creado por Facebook.
- Angular fue creado y es mantenido por Google.
- En Vue.js prima la ligereza y la velocidad de ejecución.
- Los scripts nacieron como fragmentos de código que realizaban ciertas tareas o rutinas concretas.
- En JavaScript se puede colocar el código dentro y fuera del HTML, aunque se aconseja colocarlo fuera.

- Entre las posibilidades que ofrece JavaScript, están:
 - Modificar el contenido de una página web.
 - Cambiar atributos de objetos HTML.
 - Cambiar el estilo CSS.
- JavaScript se puede comunicar con el exterior mediante:
 - Console.log().
 - Mediante el atributo innerHTML.
 - document.write().
 - window.alert().

Ejercicios propuestos



1. Crea un segundo botón para el ejemplo del apartado 1.6.1 que modifique el título "Modificando el contenido".
2. Elabora una secuencia de imágenes con al menos 4 fotogramas. Según el usuario vaya haciendo clic sobre las imágenes, la secuencia se irá reproduciendo.
3. Realiza un test de siete preguntas donde cada pregunta tenga dos botones (verdadero y falso). Dependiendo de si el usuario hace clic en verdadero o en falso la frase se pone verde (acierto) o roja (error).
4. Desarrolla una página web en la que al pulsar un botón se genere un mensaje por consola.
5. Diseña una web con tres botones para que al pulsarlos se genere un mensaje de bienvenida en ruso, español e inglés en la misma página. El mensaje irá en un párrafo (etiqueta <p>). Se pide que los mensajes tengan un color diferente dependiendo del idioma.
6. Modifica el código anterior para que los mensajes de bienvenida se generen con alertas.
7. Cambia el código anterior para que los mensajes de bienvenida se generen en consola.
8. Transforma el código anterior para que los mensajes de bienvenida se generen con el método write().
9. ¿En qué se diferencia JavaScript de Java?
10. Describe las ventajas más importantes de usar JavaScript.