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.

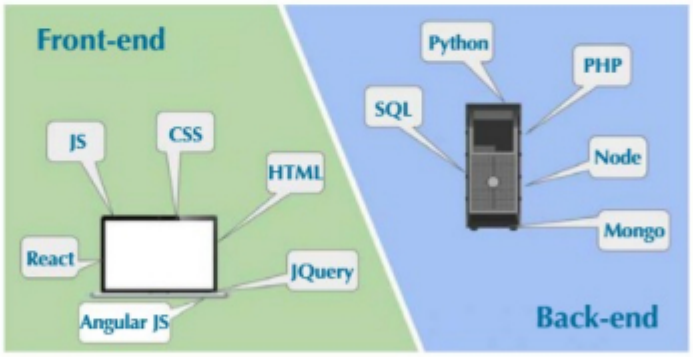

Figura 1.1
Front-end y back-end.

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.

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 ámpliamente 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.

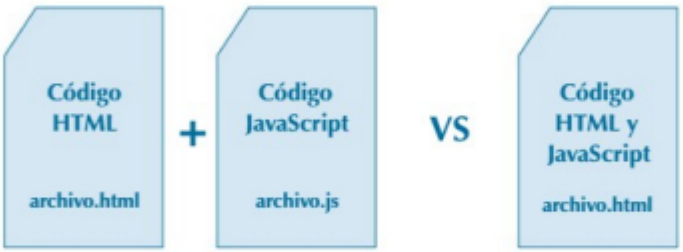

Figura 1.2
Código JavaScript dentro y fuera de un archivo HTML.

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 navegante.

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.