

# Arquitecturas y lenguajes de programación en clientes web

## 1. Lenguajes de programación en clientes web

Cuando hablamos de tecnologías empleadas en lenguajes de programación web podemos citar dos grupos básicos: client-side y server-side. Las tecnologías client-side son aquellas que son ejecutadas en el cliente, generalmente en el contexto del navegador web. Cuando los programas o tecnologías son ejecutadas o interpretadas por el servidor estamos hablando de programación server-side. Uno de los objetivos en la programación web es saber escoger la tecnología correcta para tu trabajo. Muchas veces los desarrolladores escogen rápidamente una tecnología favorita, que puede ser JavaScript, ColdFusion o PHP (Preprocesador Hipertexto) y la usan en todas las situaciones. La realidad es que cada tecnología tiene sus pros y sus contras. En general las tecnologías client-side y server-side poseen características que las hacen complementarias más que adversarias. Por ejemplo, cuando añadimos un formulario para recoger información y guardarla en una base de datos, es obvio que tendría más sentido chequear el formulario en el lado del cliente para asegurarnos que la información introducida es correcta, justo antes de enviar la información a la base de datos del servidor. La programación en el lado del cliente consigue que la validación del formulario sea mucho más afectiva y que el usuario se sienta menos frustrado al cubrir los datos en el formulario. Por otro lado, el almacenar los datos en el servidor estaría mucho mejor gestionado por una tecnología del lado del servidor (server-side), dando por supuesto que la base de datos estará en el lado del servidor. El lenguaje cliente principal es HTML (lenguaje de marcado de hipertexto, HyperText Markup Language), ya que la mayoría de páginas del servidor son codificadas siguiendo este lenguaje para describir la estructura y el contenido de una página en forma de texto. Existen algunas alternativas y variaciones de este lenguaje tales como XML (lenguaje de marcas extensible, eXtensible Markup Language), DHTML (Dynamic HTML) o XHTML (eXtensible HTML). Con el fin de mejorar la interactividad con el usuario, en este grupo de lenguajes cliente podemos incluir todos los lenguajes de script, tales como JavaScript (el más utilizado dentro de esta rama) o VBScript. También existen otros lenguajes más independientes, como ActionScript (para crear contenido Flash) o AJAX (como extensión a JavaScript para comunicación asíncrona). Cada tipo general de programación tiene su propio lugar y la mezcla es generalmente la mejor solución. Cuando hablamos de lenguajes de programación en clientes web, podemos distinguir dos variantes:

- Lenguajes que nos permiten dar formato y estilo a una página web (HTML, CSS, etc).
- Lenguajes que nos permiten aportar dinamismo a páginas web (lenguajes de scripting). En este módulo nos vamos a centrar principalmente en estos últimos, los lenguajes de scripting, y en particular en el lenguaje JavaScript que será el lenguaje que utilizaremos a lo largo de todo este módulo formativo. Hemos escogido JavaScript porque es el lenguaje de script más utilizado en la programación en el lado del cliente, y está soportado mayoritariamente por todas las plataformas. Por lo tanto a partir de ahora todas las referencias que hagamos estarán enfocadas hacia JavaScript.

Javascript, originariamente llamado LiveScript, es un lenguaje de programación creado por Netscape y nació en 1995 con un propósito muy claro: **Mejorar la navegación del usuario** directamente desde el navegador.

## 2. JavaScript

Javascript es la tercera pieza fundamental del desarrollo web frontend, junto con los lenguajes HTML y CSS. Cada uno de estos tres lenguajes tiene una función muy concreta en el desarrollo web:

- El HTML se utiliza para conformar el esqueleto y la estructura de los contenidos de una página web.
- El CSS define el estilo y la apariencia web.
- Javascript rompe con la estaticidad del HTML y permite crear elementos dinámicos e interactivos, mejorando ampliamente la interacción de los usuarios con una página web.

### 2.1 Características de Javascript

**Por** ser un lenguaje **interpretado**, no necesita de compilación y, **por** lo tanto, no utiliza los recursos del servidor. **JavasCript** es un lenguaje que soporta la programación orientada a objetos.

#### Lenguaje del lado del cliente:

Cuando se dice que un lenguaje es del lado del cliente, nos referimos a que se ejecuta en la máquina del propio cliente a través de un navegador. Algunos de estos lenguajes son el propio javascript, HTML, CSS o Java.

Esta categoría de lenguajes se diferencia de la otra gran categoría: los lenguajes del lado del servidor. Estos lenguajes se ejecutan e interpretan por el propio servidor y necesitan un tratamiento antes de mostrarlos al usuario final. Algunos de los lenguajes de programación del lado del servidor más conocidos son PHP, ASP o PERL.

#### Lenguaje orientado a objetos:

Javascript es un lenguaje orientado a objetos. Que un lenguaje esté orientado a objetos quiere decir que utiliza clases y objetos como estructuras que permiten organizarse de forma simple y son reutilizables durante todo el desarrollo. Otros lenguajes orientados a objetos son Java, Python o C++.

#### De tipado débil o no tipado:

Que un lenguaje sea de tipado débil quiere decir que no es necesario especificar el tipo de dato al declarar una variable. Esta característica supone una gran ventaja a la hora de ganar rapidez programando, pero puede provocar que cometamos más errores que si tuviéramos esa restricción que poseen los lenguajes de tipado fuerte como C++ o Java.

#### **De alto nivel:**

Que Javascript sea un lenguaje de alto nivel significa que su sintaxis es fácilmente comprensible por su similitud al lenguaje de las personas. Se le llama de “alto nivel” porque su sintaxis se encuentra alejada del nivel máquina, es decir, del código que procesa una computadora para ejecutar lo que nosotros programamos.

Un **lenguaje de alto nivel** como Javascript permite que su barrera de entrada y su curva de aprendizaje se acorte drásticamente. Un ejemplo podría ser que la sentencia condicional empiece por “IF” que significa “si...” en inglés, permitiendo asociar rápidamente su funcionamiento y significado. Otro lenguaje de alto nivel muy utilizado y uno de los mejores para iniciarse en programación por esta característica es Python.

#### **Lenguaje interpretado:**

Javascript es un lenguaje interpretado porque utiliza un intérprete que permite convertir las líneas de código en el lenguaje de la máquina. Esto tiene un gran número de ventajas como la reducción del procesamiento en servidores web al ejecutarse directamente en el navegador del usuario, o que es apto para múltiples plataformas permitiendo usar el mismo código.

Además de JS, otros ejemplos de lenguajes **interpretados son C#, Ruby, Java o Python.**

A la hora de elegir si aprender o no un nuevo lenguaje, no sólo hay que informarse sobre el tipo de lenguaje o su curva de aprendizaje, si no también su demanda en el mercado. Javascript es en la actualidad uno de los lenguajes más demandados de los últimos años por su versatilidad y su infinita capacidad para crear plataformas cada vez más atractivas.

Según un estudio de requisitos solicitados en las ofertas de empleo en el año 2020 realizado por la universidad de Boston Northeastern, Javascript es el segundo lenguaje más demandado sólo por detrás de Python.

#### **Librerías de Javascript:**

Javascript posee un amplio repertorio de librerías para diversas funcionalidades gracias a su amplia comunidad de desarrollo. A continuación, incluimos algunas de las librerías o frameworks más utilizadas en el mercado y cuales son sus características principales:

##### **jQuery**

La librería jQuery es una de las librerías más conocidas para programar en javascript, y cuenta con una gran comunidad de usuarios y desarrolladores. Una de sus principales características es que se trata de una librería open source, es decir, de código abierto. La filosofía de jQuery se basa en realizar órdenes de codificación simples y escuetas, programando en una o dos líneas lo que en javascript llevaría 20 líneas. Esta característica simplifica enormemente el trabajo de desarrollo, haciéndola muy popular en el sector.

Además, esta librería cuenta con una gran cantidad de extensiones o plugins que permiten añadir más funcionalidades al core, dotando al desarrollador de una gran flexibilidad y capacidades a la hora de afrontar un proyecto en javascript. Empresas tan importantes como Google, WordPress e IBM confían en jQuery para varios de sus proyectos

### **React**

Junto con jQuery, React JS es otra librería clave de Javascript para los desarrolladores web. React fue creada por Facebook en 2011 y planeada explícitamente para construir interfaces de usuario dinámicas, rápidas e interactivas. Al igual que jQuery, también es una librería de código abierto centrada en la parte frontend de una aplicación.

Ha ganado una gran popularidad en el sector debido a que se necesita menos código que si se utilizase únicamente javascript, y sus resultados y funcionalidades son excelentes. Además, su uso de VirtualDOM permite agilizar el proceso de actualización, ya que sólo afecta a los componentes modificados y no actualiza todos los componentes como en el desarrollo convencional.

### **AngularJS**

AngularJS es un framework desarrollado por Google en 2009 y de código abierto. Al igual que React, esta librería se centra en el desarrollo frontend. AngularJS utiliza una versión de Javascript llamada Typescript. Es una de las librerías más utilizadas junto con React o Vue por sus sencillas implementaciones y su multitud de herramientas, así como su integración y utilización del HTML evitando muchos quebraderos de cabeza a los desarrolladores.

### **Vue.js**

Con total seguridad, Vue es el framework de Javascript que más ha crecido en popularidad. Su flexibilidad y sencillez han convertido a Vue en la librería preferida de muchos desarrolladores amateur y profesionales al enfrentarse a un proyecto de desarrollo web. Al igual que las librerías anteriores, Vue es de código abierto y se creó en 2014 por el mismo que desarrolló el framework AngularJS en Google unos años antes. La característica principal que ha lanzado a Vue es su desacoplamiento en diferentes módulos, permitiendo agregar módulos y funcionalidades a la librería central de forma fácil e intuitiva.

### **Node js**

Node.js es la librería opensource más utilizada para el desarrollo backend con millones de desarrolladores en todo el mundo. Este framework surgió como respuesta a la necesidad de ejecutar aplicaciones con javascript no sólo en un navegador si no también en una máquina. Utiliza el entorno V8 de Chrome, haciendo al framework muy eficiente y seguro ante posibles bloqueos para aplicaciones que necesitan un flujo de datos en tiempo real.

## **2.2. Arquitectura Cliente/Servidor**

En entornos web, la configuración arquitectónica más habitual se basa en el modelo denominado Cliente/Servidor, basado en la idea de servicio, en el que el cliente es un componente consumidor de servicios y el servidor es un proceso proveedor de servicios.

Además, esta relación está robustamente cimentada en el intercambio de mensajes como el único elemento de acoplamiento entre ambos. Esta arquitectura tiene tres elementos comunes:

- El lado del servidor (server-side): incluye el hardware y software del servidor Web así como diferentes elementos de programación y tecnologías incrustadas. Las tecnologías pueden abarcar un rango amplio desde programas CGI (Interfaz Común de Entrada) hasta aplicaciones multihilo basadas en Java, incluyendo tecnologías de servidor de bases de datos que soporten múltiples sitios web.
- El lado del cliente (cliente-side): este elemento hace referencia a los navegadores web y está soportado por tecnologías como HTML, CSS y lenguajes como JavaScript, los cuales se utilizan para crear la presentación de la página o proporcionar características interactivas. Es justamente aquí donde nos vamos a centrar a lo largo de todo el módulo.
- La red: describe los diferentes elementos de conectividad utilizados para mostrar el sitio web al usuario. El entendimiento completo de todos los aspectos técnicos del medio Web, incluyendo la componente de red, es de vital importancia para llegar a ser un buen Diseñador Web.

### **Arquitectura de ejecución de navegadores web**

Para poder llevar a cabo el proceso de navegación web, cada navegador está formado por una serie de elementos y componentes determinados que conforman lo que se denomina arquitectura del navegador. A pesar de que cada navegador tiene su propia arquitectura, la gran mayoría de ellos coinciden en una serie de componentes básicos y comunes en todos ellos, es lo que llamamos arquitectura de referencia. Los componentes básicos incluidos en la arquitectura de referencia de un navegador web son:

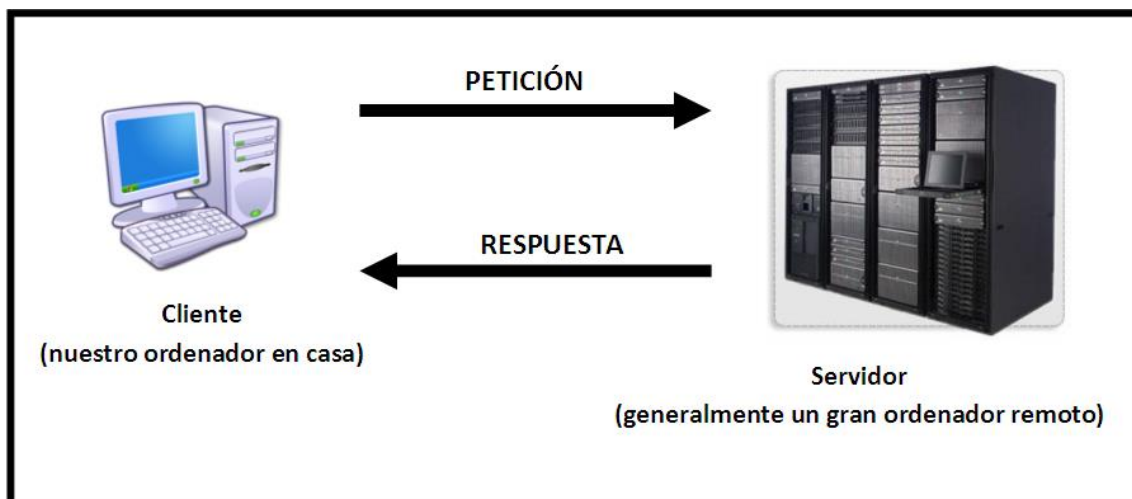
- Subsistema de interfaz de usuario. Es la capa que actúa de interfaz entre el usuario y el motor del buscador (o de navegación). Ofrece funcionalidades tales como la visualización de barras de herramientas, progreso de carga de la página, gestión inteligente de las descargas, preferencias de configuración de usuario o impresión.
- Subsistema del motor del buscador o motor de navegación. Su función principal es la de cargar una dirección determinada (URL o URI) y soportar los mecanismos básicos de navegación tales como ir a la página anterior o siguiente o la recarga de la página. Además, es el componente que gestiona el proceso de carga de una página (es quien le provee de información a la interfaz de usuario al respecto).
- Subsistema de renderizado. Este componente es el encargado de producir una representación visual del recurso obtenido a partir del acceso a una dirección web. El código de una página web es interpretado por este módulo. En función de los lenguajes, estándares y tecnologías soportadas por el navegador, este módulo será capaz de mostrar documentos HTML, XML, hojas de estilo CSS e incluso contenido embebido en la página (audio/vídeo) e imágenes. Además, este módulo establece las dimensiones exactas de cada elemento a mostrar y, en ocasiones, es el responsable de posicionar dichos elementos en una página.
- Subsistema de comunicaciones. Es el subsistema encargado de implementar los protocolos de transferencia de ficheros y documentos utilizados en Internet (HTTP, FTP, etc.).

- **Intérprete o motor de JavaScript.** Las páginas HTML habitualmente llevan código intercalado para la provisión de ciertas funcionalidades al usuario como puede ser la respuesta a ciertos eventos del ratón o del teclado. El lenguaje comúnmente aceptado para la programación de este código embebido es JavaScript (siguiendo el estándar ECMAScript). El intérprete de JavaScript será el encargado de analizar y ejecutar dicho código. Los motores JavaScript son exclusivos de cada navegador, y constituyen un elemento fundamental para la velocidad a la que cada browser es capaz de interpretar las instrucciones y realizar el renderizado de la página a cargar. La combinación “Layout Engine” + “Javascript Engine” es la que determina la velocidad a la que cada navegador carga las páginas web. Los JavaScript Engines adquirieron importancia a raíz de la aparición de Google Chrome en 2008, que literalmente barrió a la competencia en velocidad de ejecución. Pronto Mozilla y WebKit (Safari) reaccionaron, entablándose una auténtica Carrera.
- **Parser XML/JSON.** Los navegadores web suelen incluir un módulo (parser) que permite cargar en memoria una representación en árbol (árbol DOM, Document Object Model) de la página. De esta forma, el acceso a los diferentes elementos de una página por parte del navegador es mucho más rápido.
- **Componente de visualización.** Ofrece primitivas de dibujo y posicionamiento en una ventana, un conjunto de componentes visuales predefinidos (widgets) y un conjunto de fuentes tipográficas a los subsistemas principales del navegador web. Suele estar muy relacionado con las librerías de visualización del sistema operativo.
- **Subsistema de persistencia de datos.** Funciona como almacén de diferentes tipos de datos para los principales subsistemas del navegador. Estos datos suelen estar relacionados con el almacenamiento de historiales de navegación y el mantenimiento de sesiones de usuario en disco.

### Modelo de Ejecución



Ya se ha descrito que JavaScript se ejecuta en el lado del cliente de la web, y se puede utilizar para estilizar/programar cómo se comportan las páginas web cuando ocurre un evento. JavaScript es un potente lenguaje de scripts y fácil de aprender, ampliamente utilizado para controlar el comportamiento de las páginas web.



El proceso básico es el envío de una petición (que puede llevar incorporada información como los datos de un formulario) a un servidor, esperar respuesta por parte del servidor y recibir la respuesta en nuestro computador. Cada proceso de este tipo consume tiempo, el tiempo total podríamos verlo desde el lado de nuestro computador como  $\text{Tiempo Total Proceso} = \text{tiempo envío petición} + \text{tiempo procesamiento petición} + \text{tiempo recepción respuesta}$ .

Aún con velocidades rápidas de navegación cuantos más procesos de este tipo realicemos más lenta será la navegación web. JavaScript podemos decir que supone que las respuestas del servidor sean más completas y permite que se realicen más procesos en nuestro computador (aquellos procesos que realmente pueden ser resueltos en nuestro propio computador sin necesidad de estar enviando peticiones al servidor), de modo que se reduce el número de peticiones y respuestas necesarias entre cliente y servidor.

El código JavaScript es interpretado directamente por el navegador web, sin necesidad de otros programas o procesos intermedios. Un ejemplo puede ayudarnos a comprender la idea. Supongamos que en una página web pedimos al usuario que rellene un formulario con sus datos personales, y que entre los requisitos para enviar el formulario tenemos que es obligatorio que se incluya el nombre de usuario y correo electrónico, siendo obligatorio que el nombre tenga más de una letra y que el correo electrónico contenga el carácter @ (arroba).

Supongamos que una petición y respuesta de servidor requiere de un tiempo de 2 segundos y comprobemos qué ocurriría con el control del proceso del lado del servidor o controlándolo del lado del cliente con JavaScript. Como JavaScript está en el propio computador del usuario (cliente), suponemos que los tiempos de respuesta implican 0 s de consumo de tiempo, es decir, la respuesta es inmediata.

#### CONTROL DE PROCESO DEL LADO DEL SERVIDOR

Paso	Acción del usuario	Respuesta	Tiempo navegación
1	Envía un formulario donde por error el nombre está en blanco y el correo no contiene el carácter arroba	Servidor informa de que los datos en el formulario no son válidos	2 s
2	Corrige el nombre pero se olvida de corregir el correo electrónico	Servidor informa de que los datos en el formulario no son válidos	4 s
3	Corrige el correo y todo está ok	Servidor informa que el formulario ha sido enviado correctamente	6 s

#### CONTROL DE PROCESO DEL LADO DEL CLIENTE

Paso	Acción del usuario	Respuesta	Tiempo navegación
1	Envía un formulario donde por error el nombre está en blanco y el correo no contiene el carácter arroba	Cliente detecta error e informa de que los datos en el formulario no son válidos	0 s
2	Corrige el nombre pero se olvida de corregir el correo electrónico	Cliente detecta error e informa de que los datos en el formulario no son válidos	0 s
3	Corrige el correo y todo está ok	Servidor informa que el formulario ha sido enviado correctamente	2 s

Aunque existen procesos que deben ejecutarse en el lado del servidor porque sobrecargarían al navegador del cliente, ej: cuando hay que gestionar grandes volúmenes de datos. También hay procesos que **tienen que ser realizados del lado del servidor** porque necesitan de verificaciones de seguridad que no pueden residir en el computador de un usuario. Por ejemplo, para el pago con una tarjeta de crédito es necesario que el usuario envíe el número de su tarjeta de crédito al servidor y que éste mediante un proceso seguro verifique la tarjeta y el pago. Sería disparatado pensar en enviar los números de tarjetas de crédito válidas al computador del usuario y que el proceso tuviera lugar en el computador cliente para luego informar al servidor de que el pago es correcto.