

UT01 Arquitecturas y lenguajes de programación en clientes web

La WWW concebida y creada por Tim Berners-Lee, trabajador del CERN.

¿Qué hace posible el funcionamiento de la Web?

- Componentes físicos de Internet (hubs, repetidores, puentes...).
- Protocolos de comunicaciones (TCP, IP, HTTP...).
- Sistema de nombres de dominio (DNS) para búsqueda y recuperación de recursos.
- Software específico para proveer y consumir dichos recursos.

Características:

Los navegadores van desde los que funcionan en modo texto y muestran solo el código html a los gráficos, con gran variedad de interfaces de usuario y protocolos.

Los principales navegadores permiten abrir varias páginas al mismo tiempo.

Incluyen bloqueadores de ventanas emergentes.

Pueden guardar una lista de páginas (marcadores) y usar canales web (feeds).

Pueden aumentar funcionalidades a través de complementos.

Historia:

1990: Primer navegador web **WorldWideWeb**, inventado por Tim Berners-Lee, director del World Wide Web Consortium (W3C) y más tarde renombrado a **Nexus**.

Nicola Pellow desarrolló el primer navegador web multi-plataforma **Line Mode Browser**.

Marc Andreessen crea el primer navegador web con interfaz gráfica, **Mosaic**.

Andreessen lanza **Netscape Navigator**, que se convirtió en el navegador más popular.

Microsoft lanza **Internet Explorer** basado en Mosaic y que alcanzaría el 95% del mercado.

Debuta el navegador **Opera**.

Netscape lanzó lo que se convertiría en la **Fundación Mozilla** usando software de código abierto.

Apple lanza **Safari**, alcanzando el 4%.

Mozilla lanza **Firefox 1.0**, con el 12%.

Google lanza **Chrome**, siendo el más popular con el 59%.

Microsoft reemplaza Microsoft Explorer por **Edge** con un 5%.

Interfaz de usuario:

Botones de avance, retroceso, actualización e inicio para volver a la página de inicio.

Barra de direcciones para introducir la URI (URL + URN) de un recurso deseado y mostrarlo.

Barra de búsqueda para introducir términos en un motor de búsqueda web.

Barra de estado para mostrar el progreso de carga.

Capacidad de zoom de página.

Posibilidad de ver el código HTML.

Búsqueda en la página web.

Privacidad y seguridad:

La mayoría de los navegadores son compatibles con HTTPS.

Ofrecen formas rápidas y sencillas de eliminar información personal

- Caché web
- Historial de descargas
- Historial de formularios y búsquedas
- Cookies
- Historia de navegación

Algunos tienen un modo de navegación que no almacena datos personales.

Soporte de estándares:

Los primeros navegadores web solo admitían una versión muy simple de HTML.

El rápido desarrollo de los navegadores web propietarios llevó al desarrollo de dialectos no estándar de HTML, esto produjo problemas de compatibilidad.

Los navegadores web modernos soportan una combinación de HTML y XHTML basados en estándares de la W3C y deberían ser procesados de la misma manera en todos. (A diferencia del HTML, el XHTML es un lenguaje de etiquetado basado en XML).

Para ayudarnos a comprobar qué funcionalidades de JS, CSS y HTML4 están disponibles en cada navegador se puede recurrir a caniuse.com.

Comparativa de navegadores web:

Plataformas de ejecución: Safari para Apple, Edge para el SO Windows.

Características del navegador: marcadores, gestores de descarga, etc.

Personalización de la interfaz: pestañas, bloqueadores de ventanas emergentes, lector PDF...

Soporte de tecnologías Web: CSS, JS, Java, RSS o Atom, XHTML, etc.

Licencia de software:

- **Libres:** Mozilla Firefox (licencia GNU GPL) o Google Chrome (licencia BSD).
- **Propietarias:** Edge o Safari.

Arquitectura de Ejecución

Cada navegador web dependiendo de su propósito y configuración visualiza en pantalla un recurso al que quiere acceder un usuario de una dirección.

Un navegador puede estar más centrado en

- Ofrecer respuestas más rápidas.
- Mostrar una respuesta más fiel al contenido del recurso obtenido.
- Priorizar aspectos de seguridad de las comunicaciones con el servidor, etc.

Cada navegador está formado por una serie de **elementos** y **componentes** determinados que conforman la **arquitectura del navegador**.

La mayoría de navegadores comparten una serie de componentes básicos y comunes llamada **arquitectura de referencia**.

Subsistema de interfaz de usuario:

Capa que actúa de interfaz entre el usuario y el motor del buscador (o de navegación).

Funcionalidades

- Visualización de barras de herramientas.
- Progreso de carga de la página.
- Gestión inteligente de las descargas.
- Preferencias de configuración de usuario o impresión.

En algunos casos puede comunicarse con el SO para el manejo de sesiones de usuario o almacenamiento de preferencias de visualización o configuración.

Subsistema del motor del buscador o motor de navegación:

Componente que ofrece una interfaz de alto nivel para el motor de renderizado (generar img).

Funciones

- Cargar una dirección determinada (URL o URI) y soportar los mecanismos básicos de navegación.
 - o Ir a la página anterior o siguiente.
 - o Recarga de la página.
 - o Etc.
- Gestionar las alertas de JS y el proceso de carga de la página.
- Consultar y administrar las preferencias de ejecución del motor de renderizado.

Subsistema de renderizado:

Componente 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, imágenes y contenido embebido.

Este módulo establece las dimensiones exactas de cada elemento a mostrar y en ocasiones es el responsable de posicionar dichos elementos en una página.

Motores de renderizado más utilizados:

Gecko (Firefox).

WebKit (motor de iOS, GNOME Web Y Konqueror).

Blink (motor actual de Chrome, Chromium, Brave, Opera y Edge).

Goanna (motor de Pale Moon, Basilisk y K-Meleon).

Subsistema de comunicaciones:

Subsistema encargado de implementar los protocolos de transferencia de ficheros y documentos utilizados en internet (HTTP, FTP, etc).

Responsable de identificar la codificación de los datos obtenidos en función de su tipo.

Capaz de identificar si el recurso obtenido es de tipo texto, audio, video... (estándar MIME).

Puede almacenar una caché de elementos accedidos recientemente.

Intérprete de JS:

JS (estándar ECMAScript) es el lenguaje más usado para hacer páginas web dinámicas.

El intérprete de JS se encarga de analizar y ejecutar dicho código.

Se puede configurar por cuestiones de seguridad o facilidad de navegación desde el motor de navegación o el motor de renderizado.

Pueden existir subsistemas intérpretes de otros lenguajes, como applets de Java, AJAX o ActionScript.

Parser XML:

Algunos navegadores incluyen un módulo (parser) que permite cargar en memoria el DOM de la página.

Permite acceder más fácilmente a los contenidos definidos en un documento HTML.

Se consigue que el acceso a los diferentes elementos de una página por parte del navegador es mucho más rápido.

Subsistema de persistencia de datos:

Almacena diferentes tipos de datos para los principales subsistemas del navegador.

- Historial de navegación
- Sesiones de usuario
- Preferencias de configuración del navegador
- Certificados de seguridad
- Cookies

Lenguajes y Tecnologías de Programación

Los lenguajes de programación del entorno cliente se ejecutan en el navegador web del cliente

El lenguaje principal es HTML.

Los lenguajes de script permiten hacer interactivas las páginas

- JS
- VBScript
- ActionScript
- AJAX

HTML:

1980 Tim Berners-Lee, trabajador del CERN, realizó un prototipo de un nuevo sistema de hipertexto para compartir documentos.

Vio una oportunidad de unir Internet con el hipertexto, por lo que especifica HTML, escribe el primer navegador llamado WoldWideWeb desarrollado con NextStep y el primer servidor HTML llamado httpd.

El primer documento formal llamado HTML Tags (Etiquetas HTML) describía 18 elementos.

Tim consideró a HTML como una aplicación de SGML (Standard Generalized Markup Language)

Propuesto formalmente para convertirse en un estándar por la IETF (Internet Engineering Task Force).

HTML es un lenguaje de marcas para crear documentos transportables a través de internet en los que fuera posible el hipertexto.

Las primeras propuestas, HTML y HTML+ tenían etiquetas para incluir imágenes, tablas y formularios.

Ninguna consiguió convertirse en un estándar oficial.

El IETF crea un grupo de trabajo para HTML que completa **HTML 2.0**, el primer estándar oficial.

Los estándares HTML los publica otro organismo de estandarización llamado **W3C** (World Wide Web Consortium).

La versión HTML 3.2 es la primera recomendación de HTML publicada por W3C.

Más adelante se encargará el W3C de mantener las especificaciones HTML con el aporte de proveedores de software comerciales como Netscape o Microsoft.

Se publica **HTML 4.0** y más adelante se publica una versión corregida con características nuevas importantes.

- Hojas de estilos CSS
- Posibilidad de añadir pequeños programas o scripts en las páginas web.
- Formularios más potentes.
- Mejora de la accesibilidad.

HTML 4.01 no añadió ninguna novedad importante.

Después el grupo de trabajo de HTML de la W3C trabajará en la versión basada en XML **XHTML**

Comienza el desarrollo de HTML5 por el Web Hypertext Application Technology Working Group (**WHATWG**) formada por empresas como Apple, Mozilla y Opera que se unieron a la **W3C** en el 2008.

Se completa y estandariza en 2014.

Hojas de estilo CSS:

La W3C propuso la creación de un lenguaje que se utilizara para describir la presentación de documentos HTML o XML.

Se presentaron 9 propuestas.

Las 2 que se tuvieron en cuenta fueron CHSS (Cascading HTML Style Sheets) de Hakon Wium Lie y la SSP (Stream-based Style Sheet Proposal) de Bert Bos.

Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (Cascading Style Sheets).

CSS nació como un complemento a las etiquetas HTML para mejorar la presentación y aspecto.

Separaban el contenido del documento de todo el aspecto de presentación de este.

La W3C apostó por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML.

A finales de 1996, la W3C publicó la primera recomendación oficial, conocida como CSS1.

Incluía formatos de texto, párrafos, fondo, colores, márgenes y listas.

CSS2 necesitó 9 años para alcanzar el estado de recomendación.

La versión de CSS que utilizan todos los navegadores hoy en día es CSS 2.1, una revisión de CSS2 que se está elaborando.

Se añadieron características como

- Posicionamiento (capas)
- Soporte de XML
- Fuentes descargables
- Funciones de impresión
- Etc

La siguiente recomendación de CSS fue CSS3, una evolución del lenguaje de las Hojas de Estilo en Cascada dividido en módulos más pequeños y pretende ampliar la versión CSS2.

Trajo muchas novedades esperadas

- Esquinas redondeadas
- Sombras y gradientes
- Transiciones y animaciones
- Nuevos layouts como multi-columnas
- Cajas flexibles o maquetas de diseño en cuadrícula (grid layouts)

JS:

JS es un lenguaje de programación interpretado (ejecuta sin compilar, línea por línea en tiempo real) para dotar de comportamiento dinámico a las páginas web.

Cualquier navegador web actual incorpora un intérprete para código JS.

Se diseñó con una sintaxis prestada de C y Java, aunque del último no tiene relación con JS ya que tienen semánticas y propósitos diferentes.

Hereda Awk y Perl con alguna influencia de Self por su sistema de prototipado de objetos.

Basado en el concepto de objeto, pero no es un lenguaje orientado a objetos.

JS es un poco confuso para los programadores de lenguajes basados en clases como Java y C#.

Los objetos en JS utilizan **herencia basada en prototipos** y heredan propiedades directamente de otros objetos sin necesidad de que sean instancias de una misma clase.

Lenguaje **débilmente tipado**, las variables pueden contener distintos tipos en diferentes momentos del programa.

Muchos errores no aparecen hasta que se ejecuta el programa.

Todas sus variantes por defecto son globales, se ubican en un espacio de nombres común denominada global object.

94,5% de las páginas utilizan JS incluyendo Gmail, Google Maps, Facebook...

El uso más común es añadir comportamiento del lado del cliente en las páginas HTML, también conocido como HTML dinámico (DHTML).

Los scripts se incorporan o se incluyen desde páginas HTML e interactúan con el Modelo de objetos de documento (DOM) de la página,

También se puede construir servidores mediante Nodejs.

AJAX:

Asynchronous JavaScript And XML

No es un lenguaje de programación, es una técnica de desarrollo de aplicaciones web, permite comunicar el navegador del usuario con el servidor del segundo plano.

Con AJAX se minimizan las comunicaciones entre el cliente y el servidor, realizándose de manera asíncrona.

El fundamento de AJAX se encuentra en la utilización de un objeto específico de JS denominado **XMLHttpRequest**.

Ajax no es una tecnología en sí misma, sino que es una combinación de 4 existentes

- Lenguaje **HTML** y **CSS**
- **DOM**, como forma de organizar en árbol los contenidos de una página para acceder más fácilmente a un elemento determinado.
- El objeto **XMLHttpRequest**, que es el que tiene implementadas las operaciones necesarias para comunicarse asíncronamente con el servidor.
- **XML**, utilizado por el objeto XMLHttpRequest para recuperar e intercambiar información con el servidor.

FrameWorks:

Para realizar interfaces en páginas web grandes que trabajan **con una sola página** es mejor utilizar un framework, los más utilizados son **React**, **Angular** y **Vue**.