

# SELECCIÓN DE ARQUITECTURAS Y LENGUAJES DE PROGRAMACIÓN EN CLIENTES WEB

**UT1.- ARQUITECTURAS Y HERRAMIENTAS DE  
PROGRAMACIÓN**



Susana López Luengo

# Objetivos

- En esta unidad se presentan los conceptos necesarios para comprender la ejecución de código en el lado cliente de un sistema de información web
  - Se conocerán las características principales de los navegadores web
  - Se describirán las alternativas tecnológicas, lenguajes de programación y de marcas que se aplican en este entorno y cómo se integran en un documento web
  - Se aprovechará para elegir las herramientas de trabajo que se emplearán a lo largo del curso.
-

# Objetivos

- Conocer alternativas para navegar por la Web
  - Conocer los mecanismos de ejecución de código en un cliente web
  - Identificar tecnologías y lenguajes asociadas a los entornos web
  - Herramientas disponibles para el desarrollo web en el entorno cliente
  - Saber cómo se integra el código en las páginas HTML
-

# Introducción

- La WWW fue concebida y creada por Tim Berners-Lee trabajador del CERN en 1989
  - ¿Qué hace posible el funcionamiento de la Web?
    - Componentes físicos de Internet (hubs, repetidores, puentes, pasarelas, encaminadores, etc.)
    - Protocolos de comunicaciones (TCP, IP, HTTP, FTP, SMTP, etc.)
    - Sistema de nombres de dominio (DNS) para la búsqueda y recuperación de recursos
    - Software específico para proveer y consumir dichos recursos
-

# Evolución y características de los navegadores web

## Características

- Los navegadores web van desde los que funcionan en modo texto y muestran solo el código html a los gráficos con una gran variedad de interfaces de usuario y protocolos
  - Los principales navegadores web permiten abrir varias páginas al mismo tiempo
  - También incluyen bloqueadores de ventanas emergentes
  - Pueden guardar una lista de páginas web (marcadores) y tienen opción de usar canales web (feeds)
  - Pueden aumentar funcionalizadas a través de complementos
-

# Evolución y características de los navegadores web

## Historia

- 1990: Primer navegador web **WorldWideWeb** fue inventado en 1990 por Sir Tim Berners-Lee, director del World Wide Web Consortium (W3C) y más tarde renombrado a **Nexus**.
  - 1991 Nicola Pellow desarrolló el primer navegador web multi-plataforma **Line Mode Browser**
  - 1993 Marc Andreessen crea el primer navegador web con interfaz gráfica, **Mosaic**
  - 1994 Andreessen lanza **Netscape Navigator** que fue pronto el navegador más popular del mundo
  - 1995 Microsoft lanza **Internet Explorer** basado en Mosaic que en 2002 alcanzaría el 95 % del mercado
-

# Evolución y características de los navegadores web

## Historia

- 1996 debuta el navegador **Opera**
  - 1998, Netscape lanzá lo que se convertiría en la **Fundación Mozilla** usando el modelo de software de código abierto
  - 2003 Apple lanza **Safari**, que alcanzará al 15% de la navegación web
  - 2004 Mozilla lanza **Firefox** 1.0, hoy en día tiene el 4% del mercado
  - 2008 Google lanza **Chrome**, que es hoy el navegador más utilizado con el 64%
  - 2010 Microsoft reemplaza Microsoft Explorer por el navegador **Edge** con una cuota actual del %3
-

# Evolución y características de los navegadores web

Cuota de mercado Agosto 2022



**statcounter**  
GlobalStats

[Press Releases](#) [FAQ](#) [About](#) [Feedback](#)

Chrome	Safari	Edge	Firefox	Samsung Internet	Opera
65.52%	18.78%	4.29%	3.15%	2.85%	2.2%

Browser Market Share Worldwide - August 2022

<http://gs.statcounter.com/browser-market-share>

---



# Evolución y características de los navegadores web

## Interfaz de usuario

- Botones de avance y retroceso, actualización e inicio para volver a la página de inicio del usuario.
  - Barra de direcciones para introducir la [URI](#) de un recurso deseado y mostrarlo.
  - Una barra de búsqueda para introducir términos en un motor de búsqueda web
  - Una barra de estado para mostrar el progreso en la carga del recurso
  - Capacidad de zoom de página.
  - Posibilidad de ver el código HTML de una página
  - Búsqueda en la página web que se muestra
-

# Evolución y características de los navegadores 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
    - Historial de navegación.
  - Algunos tienen un modo de navegación que no almacena datos personales
-

# Evolución y características de los navegadores web

## Soporte de estándares

- Los primeros navegadores web sólo admitían una versión muy simple de HTML
- El rápido desarrollo de los navegadores web propietarios llevó al desarrollo de dialectos no estándar de HTML, lo que condujo a 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 por todos los navegadores.
- Para ayudarnos a comprobar qué funcionalidades de JavaScript, CSS y HTML5 están disponibles en cada navegador podemos recurrir a <http://caniuse.com>

# Evolución y características de los navegadores web

## Soporte de estándares

### CSS Flexible Box Layout Module - CR

Global 84.52% + 12.11% = 96.63%

unprefixed: 84.19% + 3.16% = 87.35%

Method of positioning elements in horizontal or vertical stacks.  
Support includes all properties prefixed with `flex`, as well as `display: flex`, `display: inline-flex`, `align-content`, `align-items`, `align-self`, `justify-content` and `order`.

Current aligned	Usage relative	Date relative	Show all	IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
							49						
							59					4.4	
					14	54	60			10.2		4.4.4	
<sup>4</sup>				11	15	55	61	10.1	47	10.3	all	56	59
					16	56	62	11	48	11			
						57	63	TP	49				
						58	64						

Notes Known issues (9) Resources (13) Feedback

Most partial support refers to supporting an **older version** of the specification or an **older syntax**.

<sup>4</sup> Partial support is due to large amount of bugs present (see known issues)

<http://caniuse.com>

# Evolución y características de los navegadores web

## Comparativa de navegadores web

- **Plataforma de ejecución:** como Safari para dispositivos apple y Edge para el Sistema Operativo Windows
  - **Características del navegador.** marcadores, gestores de descarga, almacenamiento de contraseñas etc.
  - **Personalización de la interfaz.** pestañas, bloqueadores de ventanas emergentes, lector PDF etc.
  - **Soporte de tecnologías Web:** CSS, Java, JavaScript, RSS o Atom , XHTML , etc.
  - **Licencia de software.**
    - **Libres:** Mozilla Firefox (licencia GNU GPL) o Google Chrome (licencia BSD)
    - **Propietarias:** Edge o Safari
-

# Evolución y características de los navegadores web

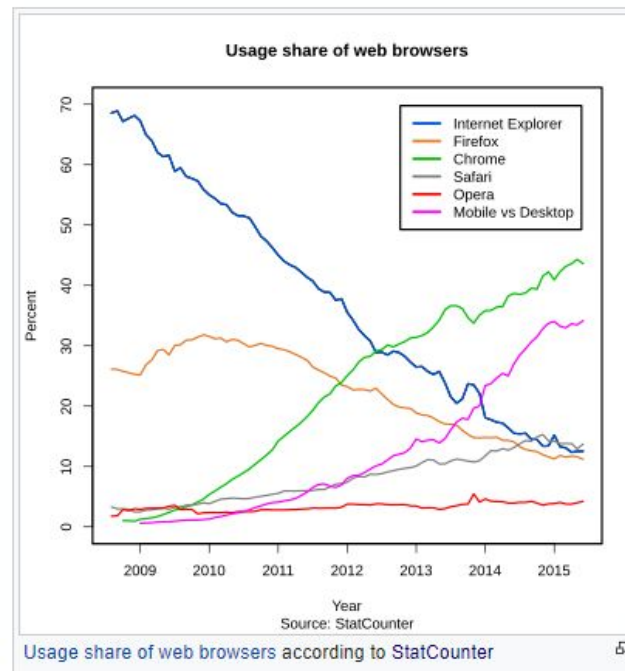
## Comparison of web browsers

From Wikipedia, the free encyclopedia

The following tables compare general and technical information for a number of web browsers.

### Contents [hide]

- 1 General information
- 2 Operating system support
- 3 Browser features
- 4 Accessibility features
- 5 Accessibility features (continued)
- 6 Acid scores
- 7 HTML5 support
- 8 Web technology support
- 9 Mobile Web technology support
- 10 Plugins and syndicated content support
- 11 JavaScript support
- 12 Protocol support
- 13 Image format support
- 14 Native multimedia support
- 15 Internationalization
- 16 Privacy
- 17 Security and vulnerabilities
- 18 See also
- 19 References



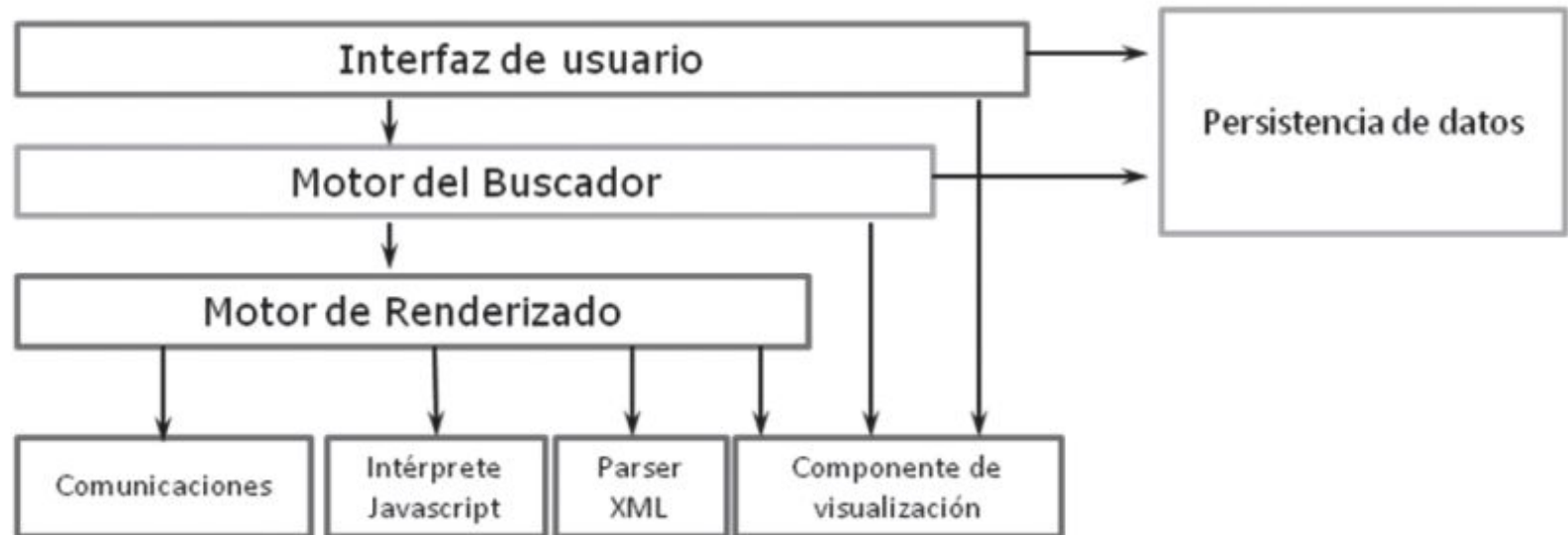
[https://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_browsers](https://en.wikipedia.org/wiki/Comparison_of_web_browsers)

# Arquitectura de Ejecución

- Cada navegador web dependiendo de su propósito y configuración visualiza en la pantalla un recurso al que quiere acceder un usuario a través de una dirección
  - Un navegador puede estar más centrado en:
    - Ofrecer una respuesta más rápida,
    - Mostrar una respuesta más fiel al contenido del recurso obtenido
    - En priorizar los aspectos de seguridad de las comunicaciones con el servidor, etc.
-

# Arquitectura de Ejecución

- 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**.



Vamos a ver cada uno...



# Arquitectura de Ejecución

## Subsistema de interfaz de usuario

- Es la 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 sistema operativo para el manejo de sesiones de usuario o el almacenamiento de preferencias de visualización o configuración.
-

# Arquitectura de Ejecución

Subsistema del motor del buscador o motor de navegación

- Es un componente que ofrece una interfaz de alto nivel para el motor de renderizado (generar imágenes)
  - Funciones
    - Cargar una dirección determinada (URL o URI) y soportar los mecanismos básicos de navegación
      - Ir a la página anterior o siguiente
      - Recarga de la página, etc...
    - Gestionar las alertas de JavaScript y el proceso de carga de una página
    - Consultar y administrar las preferencias de ejecución del motor de renderizado
-

# Arquitectura de Ejecución

## 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
-

# Arquitectura de Ejecución

## Motores de renderizado más utilizados

- **EdgeHTML**, motor de Microsoft Edge
  - **Trident**, el motor de Internet Explorer para Windows.
  - **Gecko**, utilizado en Mozilla Firefox, Mozilla Suite y otros navegadores como Galeon.
  - **Presto**, el motor de Opera 13.
  - **WebKit**, el motor de Safari, webOS y Android
  - **Blink**, motor de Chrome, Opera 14+ y Android 4.4+.
  - **Tasman**, el motor de Internet Explorer para Mac.
-

# Arquitectura de Ejecución

## Subsistema de comunicaciones

- Es el subsistema encargado de implementar los protocolos de transferencia de ficheros y documentos utilizados en Internet (HTTP, FTP, etc.).
  - Es el responsable de identificar la codificación de los datos obtenidos en función de su tipo
  - Es capaz de identificar si el recurso obtenido es de tipo texto, audio, vídeo, etc.(codificado en estándar MIME).
  - Este subsistema puede almacenar una caché de elementos accedidos recientemente
-

# Arquitectura de Ejecución

## Intérprete de JavaScript

- JavaScript (estándar ECMAScript) es el lenguaje más usado hacer páginas web dinámicas
  - El intérprete de JavaScript 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
-

# Arquitectura de Ejecución

## 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
-

# Arquitectura de Ejecución

## 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 de cliente se ejecutan en el navegador web del cliente
  - El lenguaje principal es HTML
  - Los lenguajes de script permiten hacer interactivas las páginas:
    - JavaScript
    - VBScript
    - ActionScript
    - AJAX
-

# Lenguajes y Tecnologías de Programación

## HTML

- En 1980 el físico Tim Berners-Lee, trabajador del CERN propuso y realizó un prototipo de un nuevo sistema de hipertexto para compartir documentos.
  - En 1989 Tim Berners vio la oportunidad de unir Internet con el hipertexto, por lo que especifica HTML, escribe el primer navegador llamado WorldWideWeb desarrollado con NextStep y el primer servidor HTML llamado httpd
  - El primer documento formal llamado HTML Tags (Etiquetas HTML) fue publicado a finales de 1991, este describía solamente 18 elementos que constaban su relativo diseño sencillo
-

# Lenguajes y Tecnologías de Programación

## HTML

- Berners-Lee consideró a HTML como una aplicación de SGML (Standard Generalized Markup Language)
  - Fue propuesto formalmente para convertirse en un estándar por la IETF (Internet Engineering Task Force) en 1993
  - HTML es un lenguaje de marcas para crear documentos transportables a través de Internet en los que fuera posible el hipertexto
  - Las primeras propuestas presentadas, HTML y HTML+ tenían etiquetas para incluir imágenes, tablas y formularios
  - Ninguna consiguió convertirse en un estándar oficial.
-

# Lenguajes y Tecnologías de Programación

## HTML

- A principios de 1994, el IETF crea un grupo de trabajo para HTML, que en 1995 completa **HTML 2.0**, el primer estándar oficial de HTML
  - A partir de 1996, los estándares de HTML los publica otro organismo de estandarización llamado **W3C** (World Wide Web Consortium)
  - La versión HTML 3.2 se publicó el 14 de enero de 1997 y es la primera recomendación de HTML publicada por el W3C
-

# Lenguajes y Tecnologías de Programación

## HTML

- A partir de 1996 se encargará el World Wide Web Consortium (W3C) de mantener las especificaciones HTML con el aporte de proveedores de software comerciales como Netscape y Microsoft.
  - A finales de 1997 se publica **HTML 4.0**, y a mediados de 1998 se publica una versión corregida con características nuevas muy importantes
    - Hojas de estilos CSS
    - Posibilidad de añadir pequeños programas o scripts en las páginas web
    - Formularios más potentes
    - Mejora de la accesibilidad
-

# Lenguajes y Tecnologías de Programación

## HTML

- HTML 4.01 se publica a finales de 1999 no añade ninguna novedad importante
  - Después el grupo de trabajo de HTML de la W3C trabajará en la versión basada en XML **XHTML**, que publicará su primer borrador en el año 2000
  - En 2004 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 2008
  - Se completa y estandariza el 28 de octubre de 2014.
-

# Lenguajes y Tecnologías de Programación

## 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 nueve propuestas
  - Las dos propuestas que se tuvieron en cuenta fueron la CHSS (Cascading HTML Style Sheets) de Håkon Wium Lie y la SSP (Stream-based Style Sheet Proposal) de Bert Bos
  - Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (Cascading Style Sheets).
-

# Lenguajes y Tecnologías de Programación

## Hojas de estilo CSS

- CSS nació como un complemento a las etiquetas HTML para mejorar la presentación y el aspecto de las páginas Web
  - Separaban el contenido del documento de todo el aspecto de presentación de éste
  - En 1995, la W3C decidió apostar 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, párrafo, fondo, colores, márgenes y listas
-



# Lenguajes y Tecnologías de Programación

## Hojas de estilo CSS

- CSS3, es una evolución del lenguaje de las Hojas de Estilo en Cascada dividido en módulos más pequeños y pretende ampliar la versión CSS2
  - Trajo consigo muchas novedades muy esperadas
    - Esquinas redondeadas
    - Sombras y gradientes
    - Transiciones y animaciones,
    - Nuevos layouts como multi-columnas
    - Cajas flexibles o maquetas de diseño en cuadrícula (grid layouts).
-

# Lenguajes y Tecnologías de Programación

## JavaScript

- JavaScript es un lenguaje de programación **interpretado** utilizado fundamentalmente para dotar de comportamiento dinámico a las páginas web
  - Cualquier navegador web actual incorpora un intérprete para código JavaScript
  - Se diseñó con una sintaxis prestada de C y Java, aunque del último no tiene relación con JavaScript ya que tienen semánticas y propósitos diferentes
  - También hereda de Awk y Perl con alguna influencia indirecta de Self por su sistema de prototipado de objetos.
-

# Lenguajes y Tecnologías de Programación

## JavaScript

- Está **basado en el concepto de objeto**, pero no es un lenguaje tradicional orientado a objetos
  - JavaScript es un poco confuso para programadores de lenguajes basados en clases como Java y C#
  - Los objetos en JavaScript utilizan **herencia basada en prototipos** y heredan propiedades directamente de otros objetos sin necesidad de que sean instancias de una misma clase.
-

# Lenguajes y Tecnologías de Programación

## JavaScript

- Es un lenguaje **débilmente tipado**
  - Una variable puede contener valores de distintos tipos en diferentes momentos de la ejecución del programa,
  - Muchos errores de programación no aparecen hasta que el programa es ejecutado
  - Todas sus variables por defecto son globales, se ubican en un espacio de nombres común denominado global object
-

# Lenguajes y Tecnologías de Programación

## JavaScript

- A mediados de 2017, el 94,5% de los 10 millones de páginas web más populares utilizan JavaScript entre los que incluyen Gmail, Google Maps, Facebook etc.
  - El uso más común de JavaScript 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
  - Pero también se pueden construir servidores mediante Nodejs
-

# Lenguajes y Tecnologías de Programación

## Ejemplos de JavaScript

- Galerías de imágenes: [LightBox](#)
  - Animaciones al hacer scroll: [Parallax](#)
  - Formularios: [Full Screen Form demo](#)
  - Experiencias inmersivas: [Phillippa Gregory](#)
  - Juegos: [slither.io](#)
  - Desarrollo de juegos: [phaser.io](#)
  - Servidor JavaScript: [Node.js](#)
  - Scripts para automatización: [Adobe Photoshop Scripting](#)
  - Editores: [Brackets text editor](#)
-

# Lenguajes y Tecnologías de Programación

## AJAX

- AJAX significa **Asynchronous JavaScript And XML**
  - No es un lenguaje de programación, es una técnica de desarrollo de aplicaciones web que permite comunicar el navegador del usuario con el servidor en 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 JavaScript denominado **XMLHttpRequest**
-

# Lenguajes y Tecnologías de Programación

## AJAX

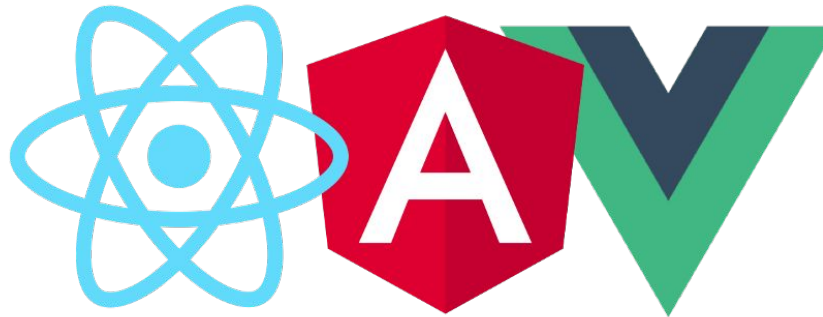
- AJAX no es una tecnología en sí misma, sino que en realidad es una combinación de 4 tecnologías existentes
    - Lenguaje **HTML** y **CSS**
    - **DOM**, como forma de organizar en árbol los contenidos de una página para así poder acceder más fácilmente a un elemento determinado
    - El objeto **XMLHttpRequest**, 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
-



# Lenguajes y Tecnologías de Programación

## FrameWorks

- Para realizar interfaces en páginas web grandes que trabajan **con una sola página** es mejor utilizar un framework
- Los más utilizados son React.js, Angular2 y Vue.js



**ESCUELA PÚBLICA:**

**DE TOD@s**

**PARA TOD@s**

*Valetecas*