



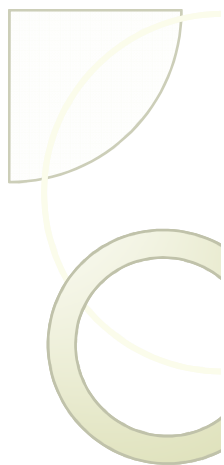
Diferencia entre Java y JavaScript

Java es un lenguaje pensado para realizar cualquier tipo de programación en cualquier ámbito, desde la programación de aplicaciones hasta páginas de servidor, sistemas distribuidos, etc. Mientras que **JavaScript** sirve tan sólo para crear scripts que se ejecuten en el navegador del usuario cuando visita páginas web.

JavaScript **NO** tiene nada que ver con Java, salvo sus orígenes. Su único parecido es el nombre y algunos aspectos de sintaxis, ya que su finalidad y filosofía son muy distintos.

Diferencia entre Java y JavaScript

	Java	JavaScript
Compilador	Necesita un kit de desarrollo y un compilador.	No se compila, sino que se interpreta por parte del navegador.
Orientado a objetos	Es orientado a objetos.	No está orientado a objetos, por lo que no se necesitan crear clases. Se realiza como un programa estructurado.
Propósito	Lenguaje de propósito general.	Lenguaje de programación de propósito específico (desarrollos web).
Estructuras fuertes	Lenguaje fuertemente tipado.	Lenguaje débilmente tipado.
Utilización	Su uso principal del lado del servidor.	Su uso principal del lado del cliente..
Otras características	Requiere de un kit de desarrollo y máquina virtual para poder programar en él.	No requiere nada específico para programar, únicamente un navegador web para ver los resultados y un editor de texto para escribir código.



U.T. I

SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

Desarrollo Web en Entorno Cliente
(Curso 2016 - 2017)



Objetivos (1/2)



- Conocer las diferentes alternativas existentes para la navegación web en función de las diferentes tecnologías web que se ejecutan en un cliente.
- Reconocer las capacidades de la ejecución de código en el lado del cliente de acuerdo a los componentes arquitectónicos de un navegador web.



Objetivos (2/2)



- Identificar los principales lenguajes y tecnologías de programación en entorno cliente.
- Conocer las técnicas de integración del código con documentos HTML.



Contenidos



1. Evolución y características de los navegadores web.
2. Arquitectura de ejecución.
3. Lenguajes y tecnologías de programación en entorno cliente.
 - HTML y derivados.
 - CSS.
 - JavaScript.
 - Applets de Java.
 - Ajax.
 - Flash.



Contenidos



4. Integración del código con las etiquetas html.

- Javascript en el mismo documento html.
- Javascript en un archivo externo.
- Javascript en elementos html.



I. EVOLUCIÓN Y CARACTERÍSTICAS DE LOS NAVEGADORES WEB.

- World Wide Web.

- Conjunto de recursos interconectados que conforman el conocimiento humano actual.

- ✚ *Hubs*, repetidores, puentes, pasarelas, encaminadores.

- ✚ Protocolos de comunicaciones: TCP, IP, HTTP, FTP, SMTP.

- ✚ Sistema de nombres de dominio (DNS).

- Configuración arquitectónica más habitual:
Cliente/Servidor.

- *Cliente* es un componente consumidor de servicios.

- *Servidor* es un proceso proveedor de servicios.



I. EVOLUCIÓN Y CARACTERÍSTICAS DE LOS NAVEGADORES WEB.

○ *Navegador web:*

- Componente software que se utiliza en el cliente y que permite acceder al contenido ofrecido por los servidores de Internet sin la necesidad de que el usuario instale un nuevo programa.
- Aplicación, distribuida habitualmente como software libre, que permite a un usuario acceder (y, normalmente, visualizar) a un recurso publicado por un servidor Web a través de Internet y descrito mediante una dirección URL (*Universal ResourceLocator*).



I. EVOLUCIÓN Y CARACTERÍSTICAS DE LOS NAVEGADORES WEB.

- Navegador web. Ejemplos:
 - **Mosaic.** Uno de los primeros navegadores Web y el primero con capacidades gráficas. Su desarrollo se abandonó en 1997.
 - **Netscape Navigator**(después **Communicator**). Fue el primer navegador en incluir un módulo para la ejecución de código *script* (JavaScript). Su desarrollo se abandonó en 2008.
 - **Internet Explorer.** Es el navegador de Microsoft. Su distribución ha sido muy elevada gracias a su integración con los sistemas Windows.
 - **Mozilla Firefox.** Se trata de un navegador de código abierto multiplataforma de gran aceptación.

I. EVOLUCIÓN Y CARACTERÍSTICAS DE LOS NAVEGADORES WEB.

- **Google Chrome.** Es el navegador de Google compilado a partir de componentes de código abierto. Sus características principales son la seguridad, velocidad y estabilidad.
- **Safari.** Es el navegador por defecto de los sistemas de Apple.
- **Dolphin Browser.** Específico para el sistema operativo Android, fue uno de los primeros en incluir soporte para navegación multitáctil.



I. EVOLUCIÓN Y CARACTERÍSTICAS DE LOS NAVEGADORES WEB.

Estadísticas de uso de navegadores (2008-2012)

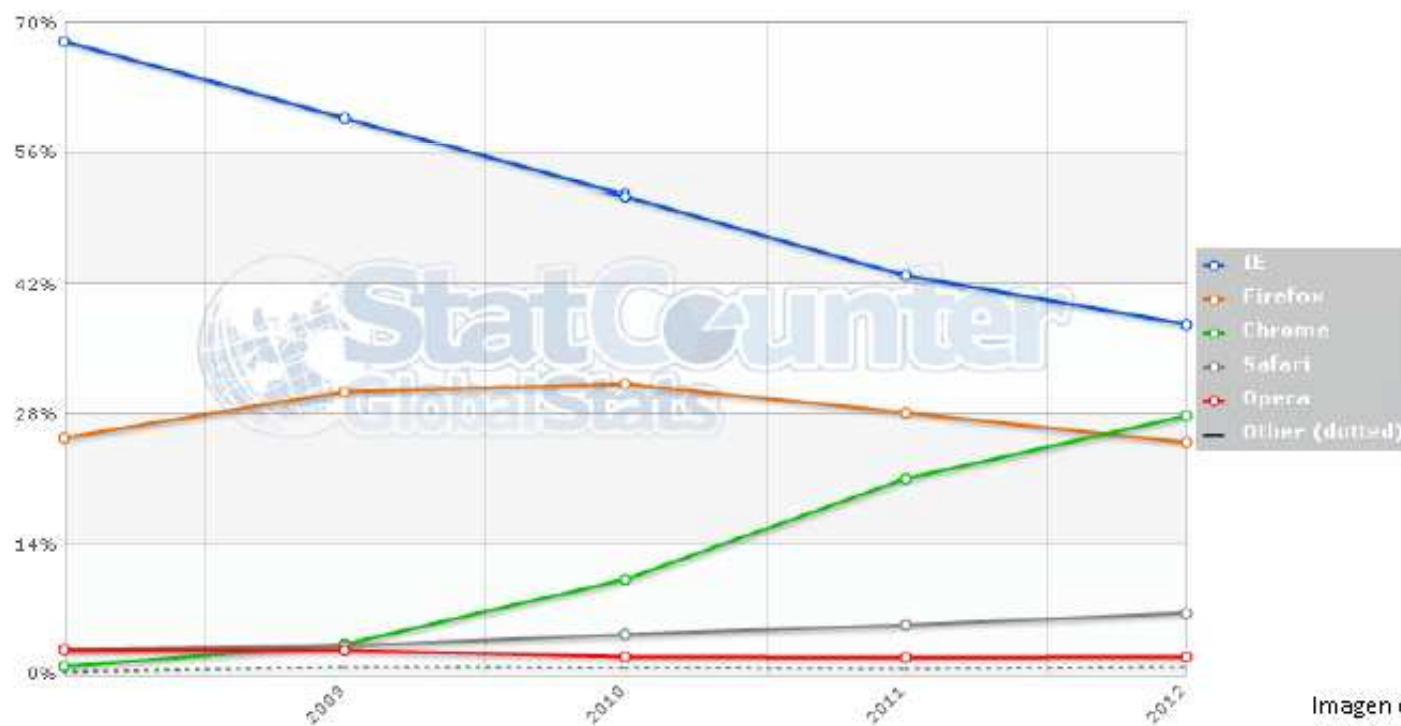


Imagen obtenida de
gs.StatCounter.com



I. EVOLUCIÓN Y CARACTERÍSTICAS DE LOS NAVEGADORES WEB.

- Navegador web. Criterios de clasificación:
 - **Plataforma de ejecución.** Sistemas Operativos.
 - **Características del navegador.** Funcionalidades adicionales.
 - **Personalización de la interfaz.** Funciones de accesibilidad.
 - **Soporte de tecnologías Web.** Grado de soporte de los estándares de la Web.
 - **Licencia de software.** Código libre y navegadores propietarios.



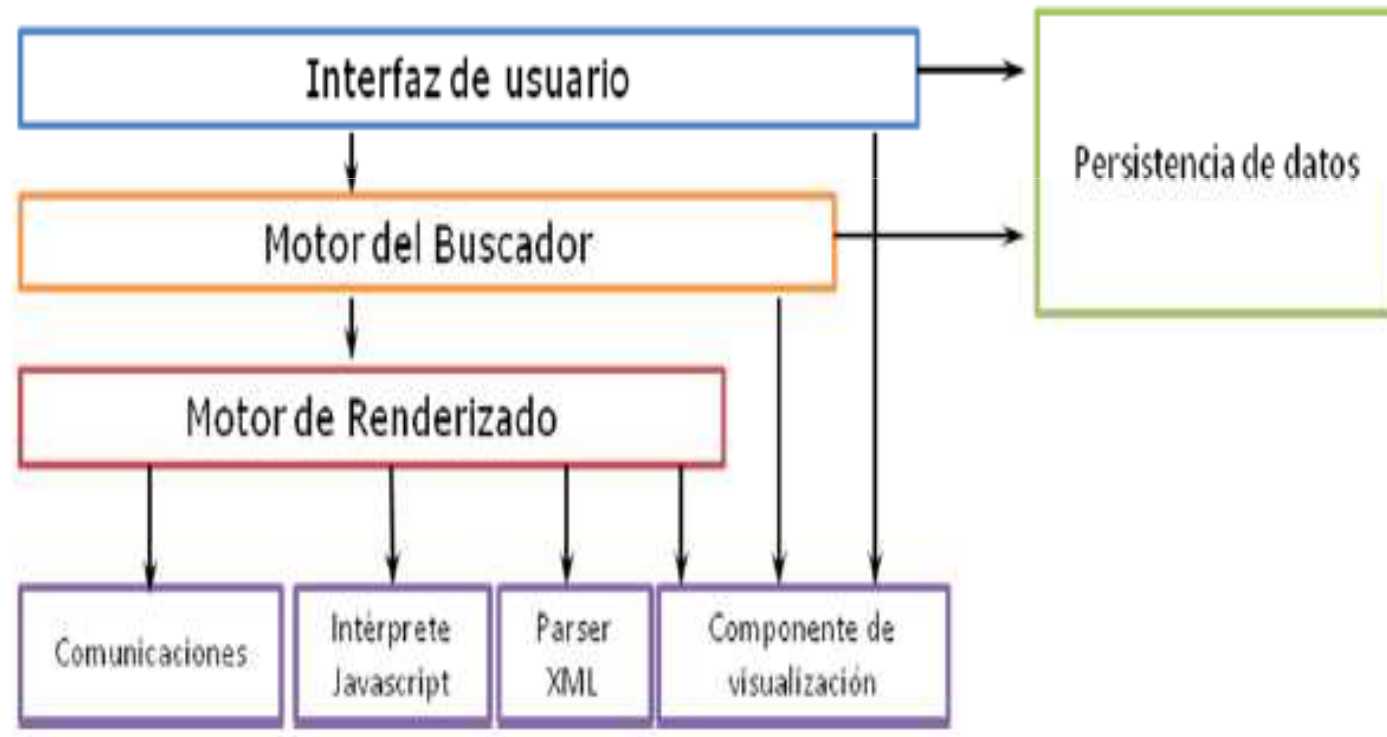
2. ARQUITECTURA DE EJECUCIÓN.

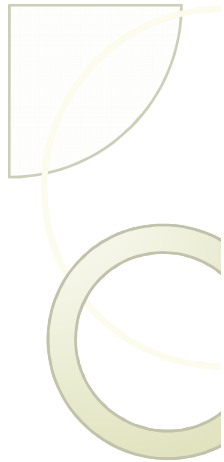
- Proceso de ejecución:
 - Se inicia con el usuario indicando la dirección del recurso al que quiere acceder y termina con la visualización del recurso por parte del navegador en la pantalla del usuario.



2. ARQUITECTURA DE EJECUCIÓN.

- **Arquitectura de referencia de un navegador web:**

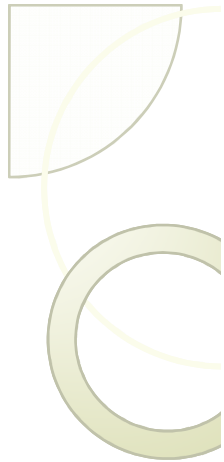




2. ARQUITECTURA DE EJECUCIÓN.

Arquitectura de referencia de un navegador web (I):

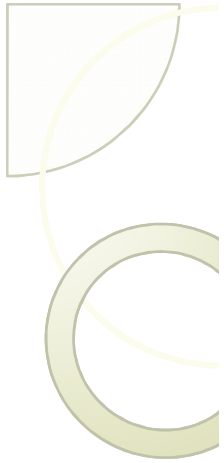
- **Subsistema de interfaz de usuario.** Es la capa que actúa de interfaz entre el usuario y el motor del buscador (o de navegación).
- **Subsistema del motor del buscador o motor de navegación.** Este subsistema es un componente que ofrece una interfaz de alto nivel para el motor de renderizado.
- **Subsistema de renderizado.** Este componente es el encargado de producir una representación visual del recurso obtenido a partir del acceso a una dirección Web.
- **Subsistema de comunicaciones.** Es el subsistema encargado de implementar los protocolos de transferencia de ficheros y documentos utilizados en Internet (HTTP, FTP, etc.).



2. ARQUITECTURA DE EJECUCIÓN.

Arquitectura de referencia de un navegador web (II):

- **Intérprete de JavaScript.** Será el encargado de analizar y ejecutar código JavaScript.
- **ParserXML.** Módulo que permite cargar en memoria una representación en árbol de la página web.
- **Componente de visualización.** Este subsistema ofrece funcionalidades relacionadas con la visualización de los contenidos de un documento HTML en una página web.
- **Subsistema de persistencia de datos.** Funciona como almacén de diferentes tipos de datos para los principales subsistemas del navegador.



3. LENGUAJES Y TECNOLOGÍAS DE PROGRAMACIÓN EN ENTORNO CLIENTE.

Los lenguajes de programación del entorno de cliente son aquellos que se ejecutan en el navegador Web.

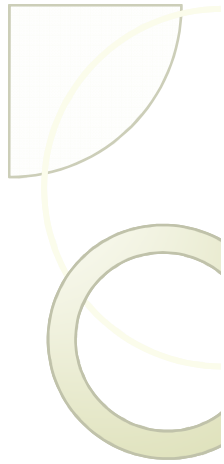
- Lenguajes principales:
 - HTML.
 - DHTML.
 - XML.
 - XHTML.
- Lenguajes de scripting:
 - JavaScript.
 - VBScript.
- Otros lenguajes:
 - ActionScript.
 - AJAX.



3. LENGUAJES Y TECNOLOGÍAS DE PROGRAMACIÓN EN ENTORNO CLIENTE.

HTML y derivados (I):

- *HTML: HyperText Markup Language* (lenguaje de marcado de hipertexto) es el lenguaje de marcas de texto más utilizado en la *World Wide Web*.
- Se basa en la utilización de un sistema de etiquetas cerrado aplicado a un documento de texto.
- No necesita ser compilado, sino que es interpretado (ejecutado a medida que se avanza por el documento HTML).
- Hipervínculo: enlace de una página web o un archivo a otra página web u otro archivo.



3. LENGUAJES Y TECNOLOGÍAS DE PROGRAMACIÓN EN ENTORNO CLIENTE.

HTML y derivados (II):

- **XML:** lenguaje de etiquetado extensible cuyo objetivo principal es describir datos para su transferencia eficiente y no mostrarlos, como es el caso de HTML.
- **XHTML:** adaptación de HTML al lenguaje XML.
- HTML Dinámico (**DHTML**): integración de HTML con lenguajes de *scripting* (JavaScript), hojas de estilo personalizadas (CSS) y la identificación de los contenidos de una página Web en formato de árbol (DOM).



3. LENGUAJES Y TECNOLOGÍAS DE PROGRAMACIÓN EN ENTORNO CLIENTE.

- **CSS (*Cascade Style Sheets*)**: sirve para separar el formato que se quiere dar a la página Web de la estructura de la página Web y las demás instrucciones.
- **JavaScript**: lenguaje de programación de *scripting* (interpretado) y, normalmente, embebido en un documento HTML.
- **Applets de Java**: pequeños componentes (objetos independientes) integrados en una página Web y programados en Java.
- **AJAX (*Asynchronous JavaScript And XML*)**: conjunto de técnicas y métodos de desarrollo Web para la creación aplicaciones Web interactivas y asíncronas.
- **Adobe Flash**: tecnología de animación actualmente bajo licencia de Adobe y que utiliza ActionScript como lenguaje principal.



4. INTEGRACIÓN DEL CÓDIGO CON LAS ETIQUETAS HTML.

- **JavaScript en el mismo documento HTML.**
 - Uso de unas etiquetas predefinidas para marcar el texto (`<script>` y `</script>`).
 - Puede incluirse en cualquier parte del documento, aunque se recomienda que se defina dentro de la cabecera del documento HTML.
 - Esta técnica suele utilizarse cuando se definen instrucciones que se referenciarán desde cualquier parte del documento o cuando se definen funciones con fragmentos de código genéricos.



Actividad 1.1



JavaScript en el mismo documento HTML.

```
<!DOCTYPE htmlPUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<htmlxmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type"
```

```
content="text/html;charset=iso-8859-1" />
```

```
<title>Ejemplo 1</title>
```

```
<script type="text/javascript">
```

```
    alert("Prueba de JavaScript");
```

```
</script>
```

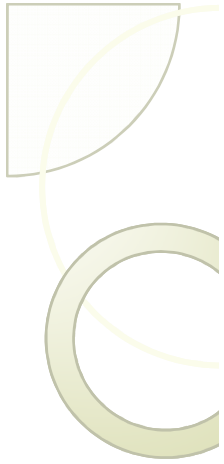
```
</head>
```

```
<body>
```

```
<h1>Ejemplo 1: código embebido</h1>
```

```
</body>
```

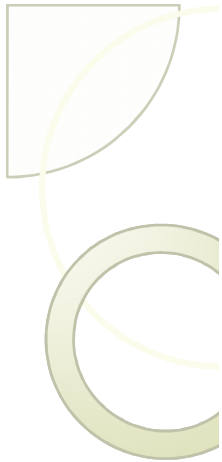
```
</html>
```



4. INTEGRACIÓN DEL CÓDIGO CON LAS ETIQUETAS HTML.

○ **JavaScript en un archivo externo.**

- Las mismas instrucciones de JavaScript que se incluyen entre un bloque `<script></script>` pueden almacenarse en un fichero externo con extensión `.js`.
- La forma de acceder y enlazar esos ficheros `.js` con el documento HTML/XHTML es a través de la propia etiqueta `<script>`.
- No existe un límite en el número de ficheros `.js` que pueden enlazarse en un mismo documento HTML/XHTML..



Actividad 1.2



○ JavaScript en un archivo externo.

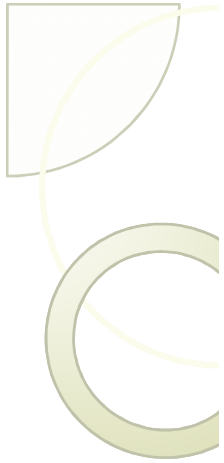
Archivo mensaje.js:

```
alert("Prueba de JavaScript");
```

Archivo ejemplo2.html:

```
<!DOCTYPE htmlPUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<htmlxmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1" />
    <title>Ejemplo 2</title>
    <script type="text/javascript"
    src="/inc/mensaje.js"></script>
  </head>
  <body>
    <h1>Ejemplo 2: fichero externo</h1>
  </body>
</html>
```



4. INTEGRACIÓN DEL CÓDIGO CON LAS ETIQUETAS HTML.

○ **JavaScript en elementos HTML.**

- Consiste en insertar fragmentos de JavaScript dentro de atributos de etiquetas HTML de la página.
- Forma de controlar los eventos que suceden asociados a un elemento HTML concreto.
- Principal desventaja: el mantenimiento y modificación del código puede resultar más complicado.



Actividad 1.3



JavaScript en elementos HTML.

```
<!DOCTYPE htmlPUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<htmlxmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1" />
    <title>Ejemplo 3</title>
  </head>
  <body>
    <p onclick="alert('Prueba de JavaScript');">
      Ejemplo 3: código en atributos
    </p>
  </body>
</html>
```