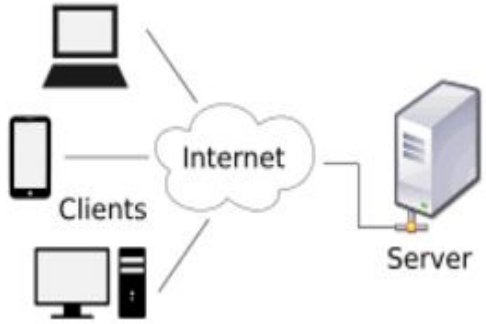
1.Selección de arquitecturas y herramientas de programación

- 1.1 Evolución y características de los navegadores web.
- 1.2 Arquitectura de ejecución.
- 1.3 Lenguajes y tecnologías de programación en entorno cliente.
- 1.4 Integración del código con las etiquetas HTML.

1.1 Evolución y características de los navegadores web

- La configuración arquitectónica de desarrollo web más habitual: modelo **Cliente/Servidor** basado en la idea de servicio.
- En este modelo, las tareas se reparten entre proveedores de servicios y recursos (servidores) y los demandantes (clientes).
- Uno de los componentes habituales en el cliente es el **navegador** web.
- **Navegador** o explorador web: aplicación que permite al usuario **acceder** a un recurso de tipo hipertexto de un servidor web a través de Internet.



1.1 Evolución y características de los navegadores web

- Exploradores de los clientes de navegación web:
  - Mosaic. Uno de los primeros y el 1° con capacidades gráficas. Base para las primeras versiones de IE y Mozilla.
  - Netscape Navigator. El 1° en incluir un módulo para ejecutar JavaScript.
  - Internet Explorer de Microsoft. Integrado en los sistemas Windows.

1.1 Evolución y características de los navegadores web

- Mozilla Firefox. Código abierto y multiplataforma. 1° en incluir navegación por pestañas.
- Google Chrome. Arquitectura multiproceso lo que permite ejecutar las pestañas independientemente.
- Safari. Navegador de Apple.
- Dolphin Browser. Específico S.O. Android. Primero en incluir soporte para navegación táctil.

1.1 Evolución y características de los navegadores web

- Criterios para diferenciar a los navegadores:
  - Plataforma de ejecución: S.Os.
  - Características del navegador: Admin. marcadores, almacenam. contraseñas, búsquedas, corrección ortográfica.
  - Personalización de la interfaz. Funciones de accesibilidad: pestañas, zoom, búsquedas avanzadas, integración de visualizadores de formatos de ficheros (pdf).

1.1 Evolución y características de los navegadores web

- Soporte de tecnologías Web. Nivel de soporte de tecnologías como CSS, Java, JavaScript, XHTML, etc.
- Licencia de software. Navegadores de código libre: Mozilla Firefox o Google Chrome y navegadores propietarios: Internet Explorer, Microsoft Edge o Safari (Apple).
- Salvo raras excepciones (OmniWeb, ya es gratuito) los más utilizados son gratuitos.

1.1 Evolución y características de los navegadores web

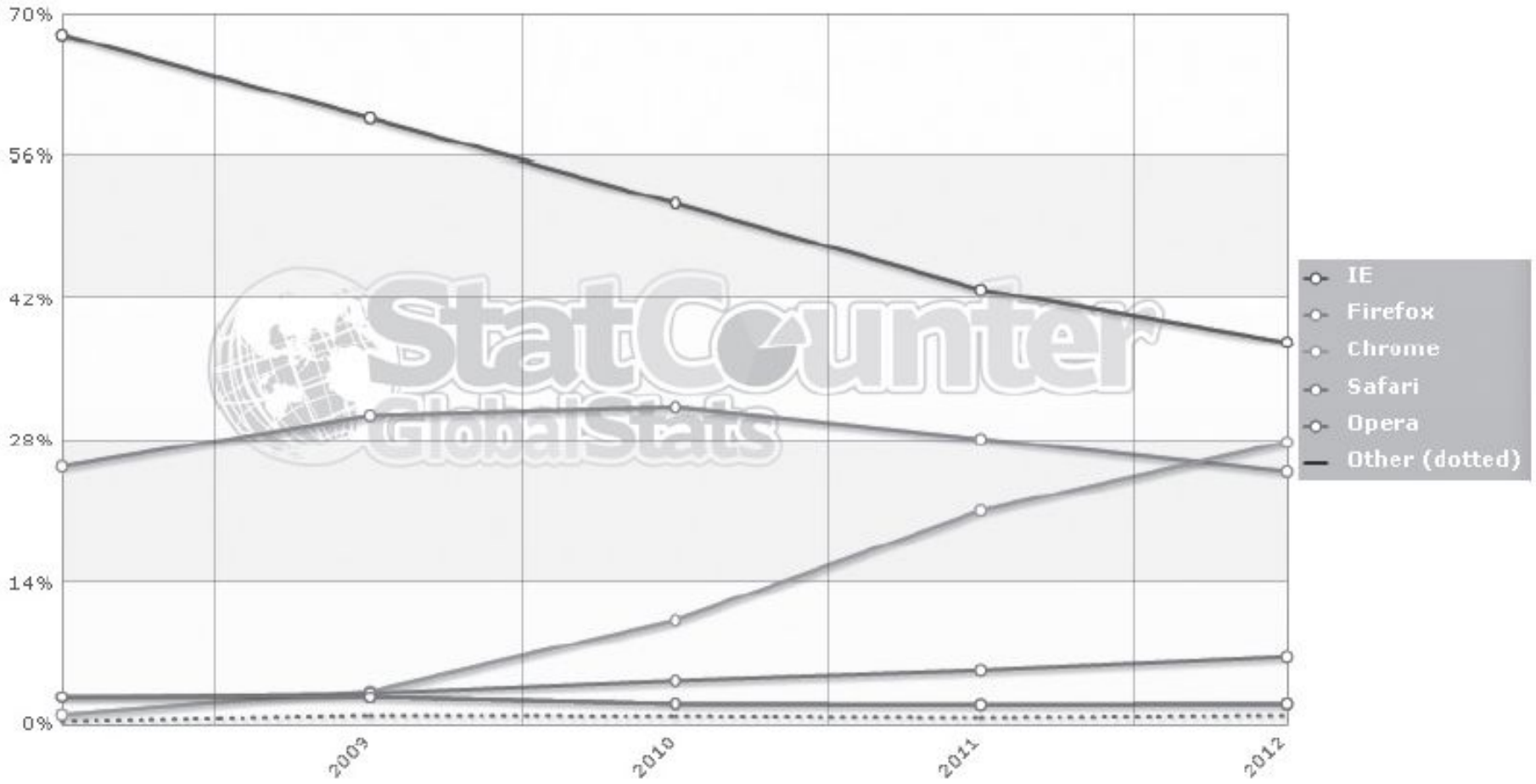
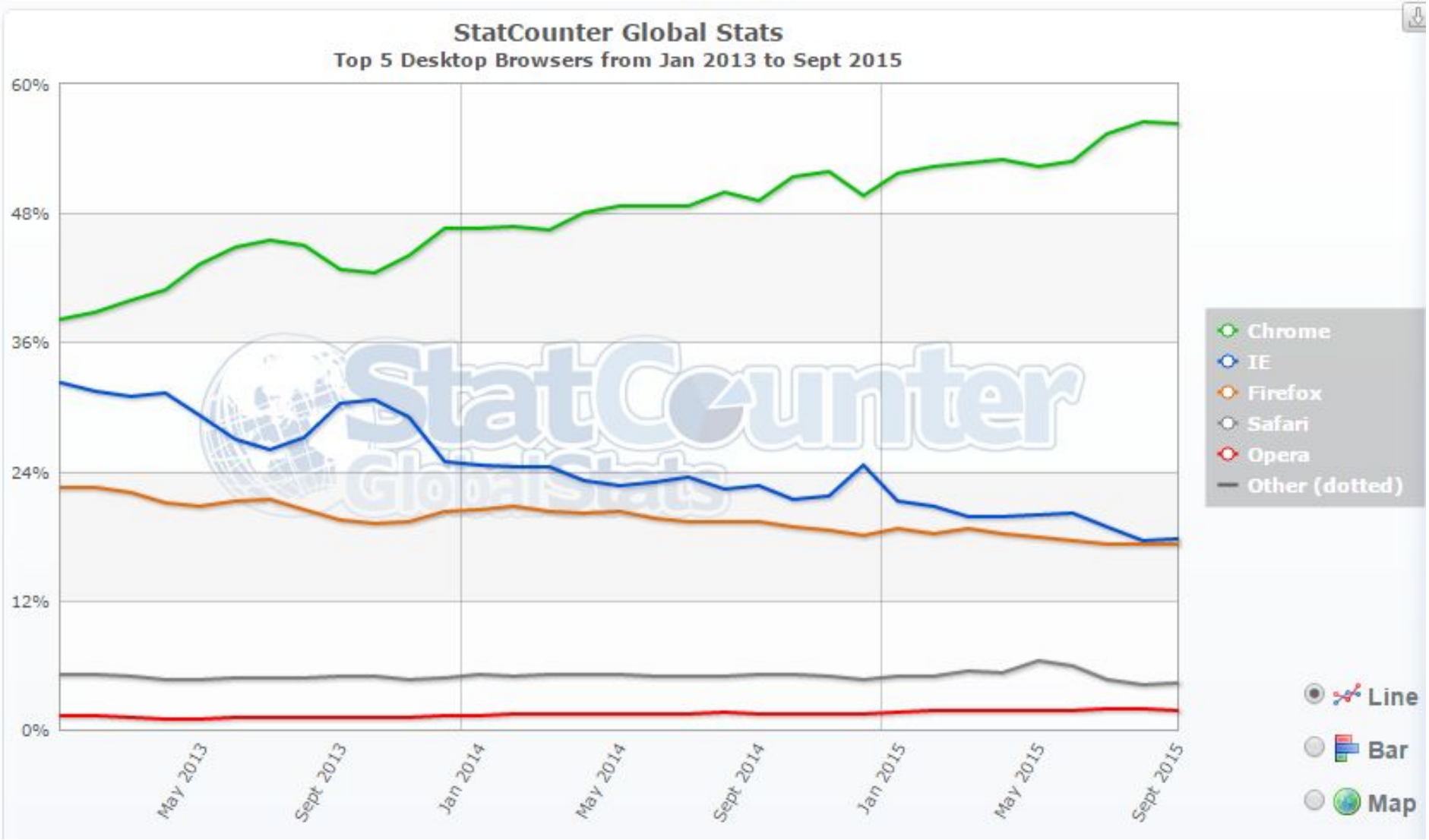
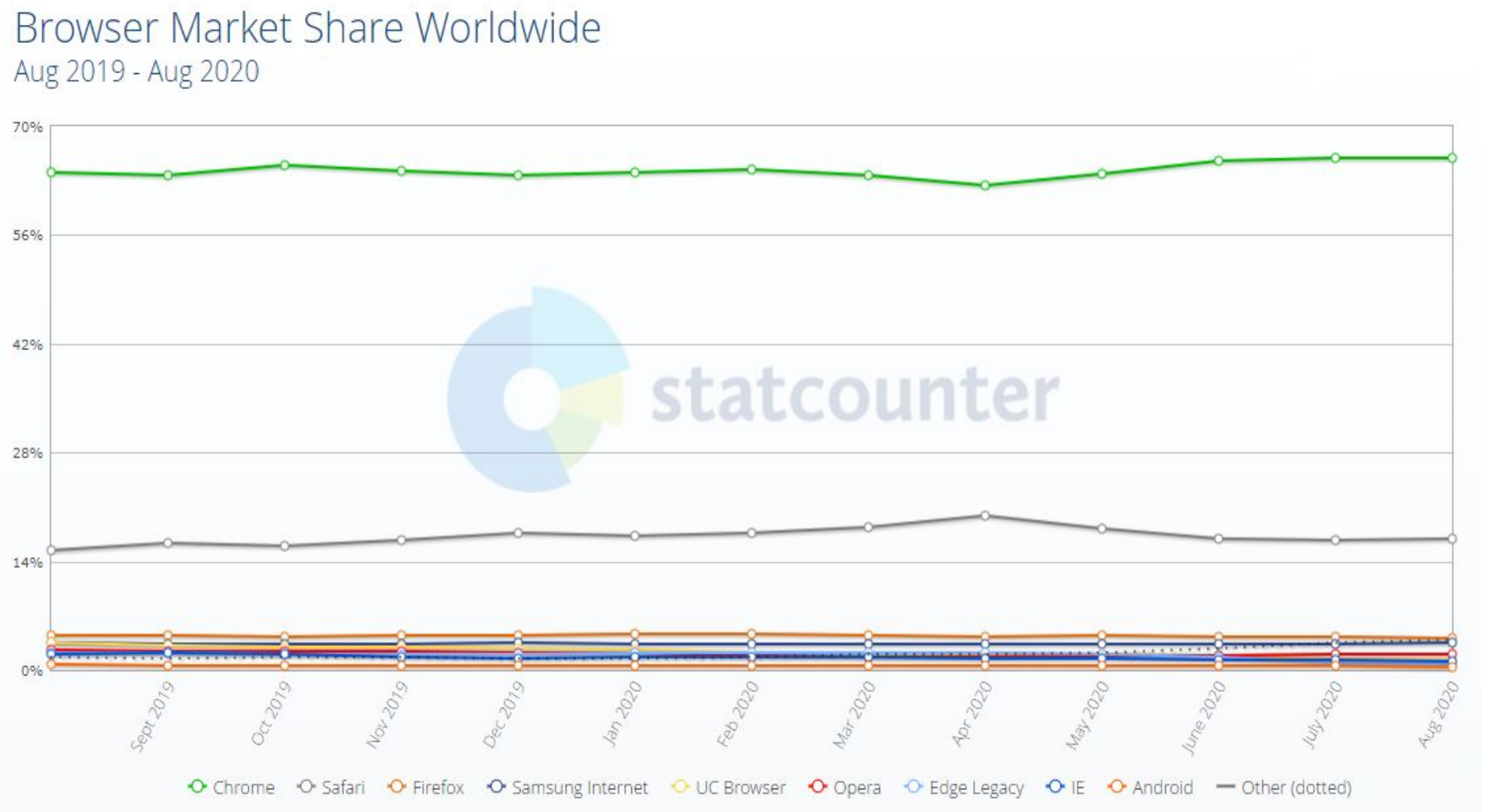


Figura 1.1. *Estadísticas de uso de navegadores (2008-2012)*

1.1 Evolución y características de los navegadores web



1.1 Evolución y características de los navegadores web



1.2 Arquitectura de ejecución

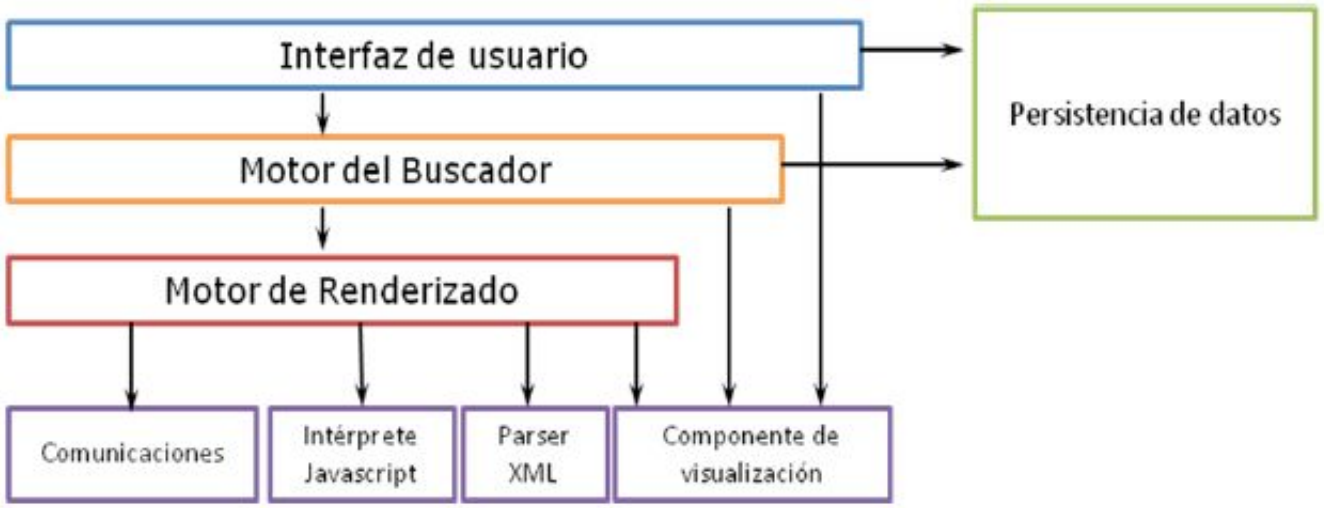
- Cada navegador web tiene su propia forma de interpretar la **interacción** con el usuario:
  - 1°: el usuario indica la dirección del recurso al que quiere acceder.
  - 2°: el navegador visualiza el recurso.

1.2 Arquitectura de ejecución

- El resultado de este proceso depende de la **configuración** y **propósito** del navegador.
- De esta forma el navegador puede estar más centrado en ofrecer:
  - Una respuesta más rápida.
  - Una respuesta más fiel al contenido del recurso obtenido.
  - Una seguridad en las comunicaciones, etc.

1.2 Arquitectura de ejecución: Elementos que intervienen

- **Interfaz de usuario**. Capa entre usuario y navegador.
- **Motor del buscador** o motor de navegación. Funciones: gestionar navegación, cargar págs, etc.
- Subsistema de **renderizado**. Encargado de producir una representación visual de la pág.
- **Comunicaciones**. Encargado de los protocolos de transferencia de ficheros y documentos (HTTP,FTP,etc).



1.2 Arquitectura de ejecución

- Intérprete de **JavaScript**. Código intercalado en las págs. HTML para responder a ciertos eventos de ratón o teclado.
- **Parser XML**. Permite cargar en memoria una representación en árbol de la página.
- **Componente de visualización**. Relacionado con las librerías de visualización del S.O.
- Subsistema de **persistencia de datos**. Almacén de diferentes tipos de datos, preferencias de configuración del navegador, certificados de seguridad, cookies, etc.

1.3 Lenguajes y tecnologías de programación en entorno cliente

- Son los lenguajes que se ejecutan en el navegador web. En una arquitectura Cliente/Servidor, en el lado del cliente.
- El lenguaje cliente principal es el HTML.
- Para mejorar la interactividad con el usuario se incluyen todos los lenguajes de script como JavaScript o VBScript.
- Otros lenguajes: ActionScript (para crear contenido Flash) o AJAX (extensión de JavaScript para comunicación asíncrona).

1.3.1 HTML y derivados.

- HTML: lenguaje de marcas de texto más utilizado en la web.
- No es un lenguaje. Se basa en un sistema de *etiquetas* cerrado y en el concepto de *hipertexto* para conectar elementos (documentos o recursos) entre sí.
- No necesita ser compilado, es interpretado.
- Su evolución ha dado lugar a XML, XHTML, DHTML.

1.3.1 HTML y derivados.

- XML: lenguaje de etiquetado extensible simple pero estricto.
- XHTML: adaptación de HTML a XML.
- DHTML: HTML dinámico. Combina HTML estático con:
  - lenguaje interpretado del lado cliente,
  - CSS y
  - DOM.

1.3.2 CSS(Cascade Style Sheets)

- Las Hojas de Estilo en Cascada sirven para **separar** el **formato** que se quiere dar a la página web de la **estructura** de la página web y demás instrucciones.

1.3.3 JavaScript

- Es un lenguaje de programación de scripting (interpretado) y normalmente embebido en un documento HTML.
- Orientado a objetos y tipado.
- Se utiliza del lado del cliente.

- Objetivo: realizar mejoras en la interfaz de usuario y crear páginas web dinámicas.
- Todos los navegadores actuales interpretan JavaScript integrado en sus páginas web.

1.3.4 Applets de Java

- Pequeños objetos independientes para funcionalidades complejas que se integran en una página web.
- Son fragmentos de código Java que se ejecutan en el cliente.
- Ventaja: son menos dependientes del navegador y del S.O. que los scripts en JavaScript.
- Desventaja: más lentos de procesar y tienen un espacio delimitado en la página donde se ejecutan (no se mezclan con todos los componentes de la página ni tienen acceso a ellos).

1.3.5 AJAX (JavaScript Asíncrono y XML)

- Conjunto de técnicas y métodos de desarrollo web para la creación de aplicaciones web interactivas.
- Se ejecutan en el cliente.
- Se mantiene una comunicación asíncrona con el servidor en segundo plano.
- Se pueden realizar cambios sobre las páginas del cliente sin necesidad de que éste tenga que recargarlas.

1.3.5 AJAX (JavaScript Asíncrono y XML)

- Es una combinación de 4 tecnologías existentes:
  - XHTML/HTML y CSS para el diseño.
  - DOM para organizar en árbol los contenidos.
  - El objeto XMLHttpRequest que tiene implementadas las operaciones necesarias para comunicarse asíncronamente con el servidor.
  - XML como lenguaje utilizado por el objeto XMLHttpRequest para recuperar o intercambiar información con el servidor.

1.3.6 Adobe Flash y ActionScript

- Flash tecnología de animación bajo licencia Adobe y que utiliza ActionScript como lenguaje principal.
- Ventaja de Flash la capacidad de crear gráficos y animaciones vectoriales. Más flexibles que las de mapas de bits y ocupan menos memoria.
- En declive por la mayor prevalencia de otras tecnologías y lenguajes como HTML 5.
- Desventajas: gran consumo de procesador y de batería en dispositivos móviles. Software 100% propietario.

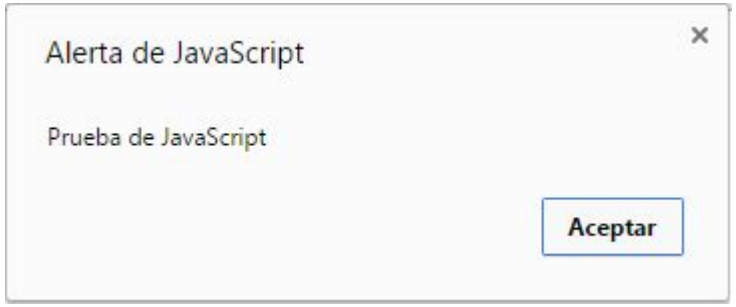
1.4 Integración del código script con las etiquetas HTML

- Para desarrollar aplicaciones web que se ejecuten en el lado del cliente, el documento base estará escrito en HTML o XHTML.
- La integración de JavaScript y HTML/XHTML se puede hacer de tres formas diferentes:
  - En el mismo documento HTML.
  - En un archivo externo.
  - En elementos HTML.

1.4.1 JavaScript en el mismo documento HTML

- Las etiquetas `<script>` y `</script>` indican la existencia de un bloque de código embebido.
- Se puede incluir en cualquier parte del documento.
- **OBSOLETO**: NO se ha de añadir el atributo `type` a la etiqueta `<script>` con el valor `text/javascript`.

1.4.1 JavaScript en el mismo documento HTML. Ejemplo




```
<html>
  <head>
   <title> Ejemplo1 </title>
   <script>
     alert("Prueba de JavaScript");
   </script>
  </head>
  <body>
     <h1>Ejemplo 1: código embebido</h1>
  </body>
</html>
```

1.4.2 JavaScript en un archivo externo.

- Las mismas instrucciones de JavaScript que se incluyen entre un bloque `<script>` y `</script>` se almacenan en un fichero externo con extensión `.js`.
- La forma de acceder y enlazar esos ficheros con el documento HTML/XHTML es a través de la propia etiqueta `<script>`.
- Además del atributo `type`, se incluye el atributo `src` que contiene la ruta relativa (con respecto al documento) del archivo JavaScript a enlazar.

1.4.2 JavaScript en un archivo externo.

- Sólo se puede enlazar un archivo en cada etiqueta `<script>` pero se pueden incluir tantas etiquetas como sean necesarias.
- Ventaja: la vinculación de un mismo fichero externo puede hacerse desde varios documentos HTML/XHTML distintos.

1.4.2 JavaScript en un archivo externo. Ejemplo

Archivo “prueba.js”:

```
alert("Prueba de archivo externo")
```

Documento html:

```
<html>
  <head>
    <title> Ejemplo2 </title>
    <script src="prueba.js"></script>
  </head>
  <body>
     <h1>Ejemplo 2: fichero externo</h1>
  </body>
</html>
```

1.4.3 JavaScript en elementos HTML

- Se suele utilizar para **controlar los eventos** que suceden asociados a un elemento HTML concreto.
- Consiste en **insertar** fragmentos de JavaScript dentro de **atributos de etiquetas HTML** de la página.
- **Desventaja:** al intercalarse el código JavaScript con el código HTML el **mantenimiento** y modificación resulta complicado.

1.4.3 JavaScript en elementos HTML. Ejemplo

```
<body>
   <p onclick="alert('Prueba de JavaScript');">
       Ejemplo 3: código en atributos
   </p>
</body>
```