

Desarrollo Web en Entorno Cliente

Tema 1

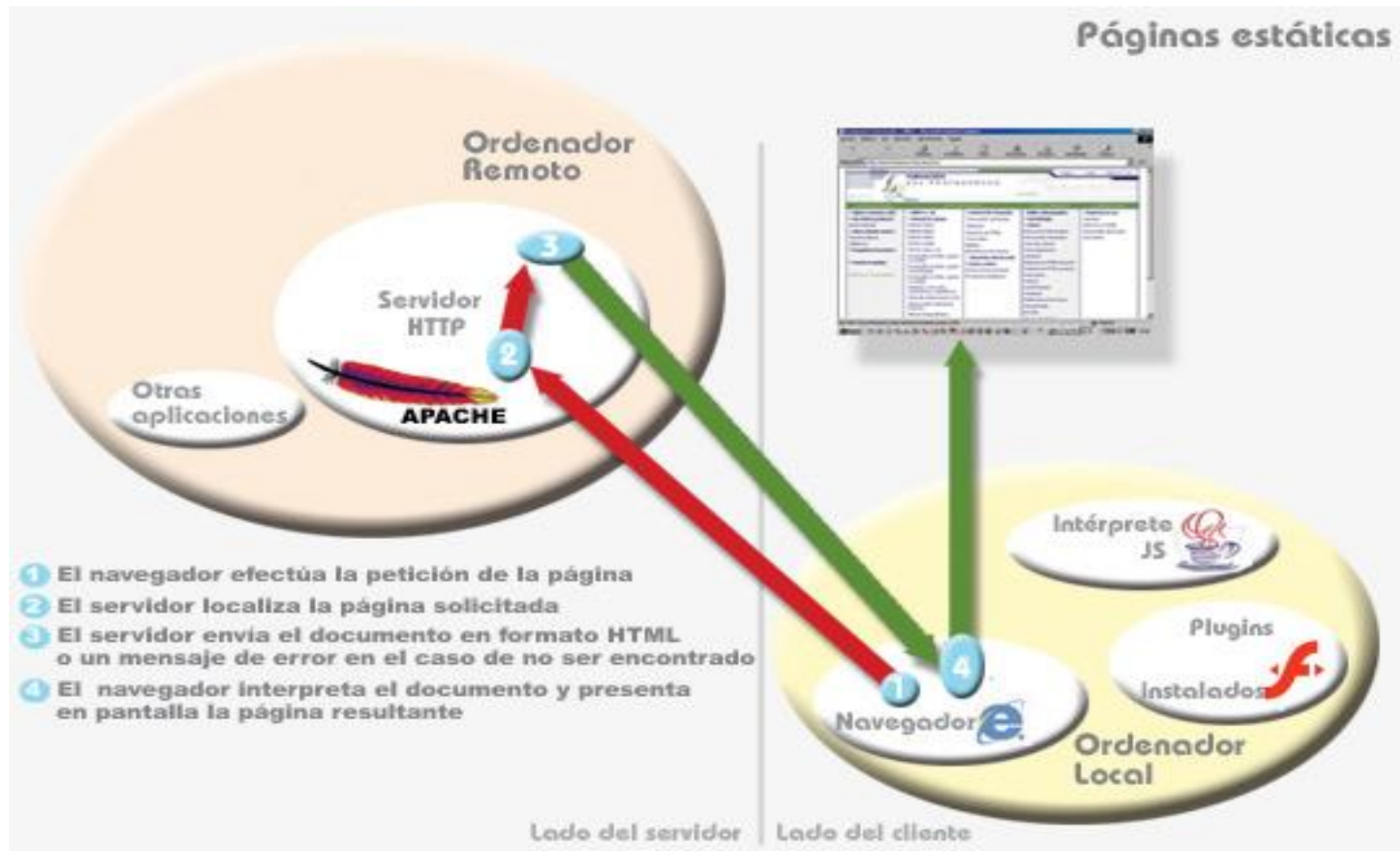
**Selección de arquitecturas y
herramientas de programación**

1. Evolución y características de los navegadores Web

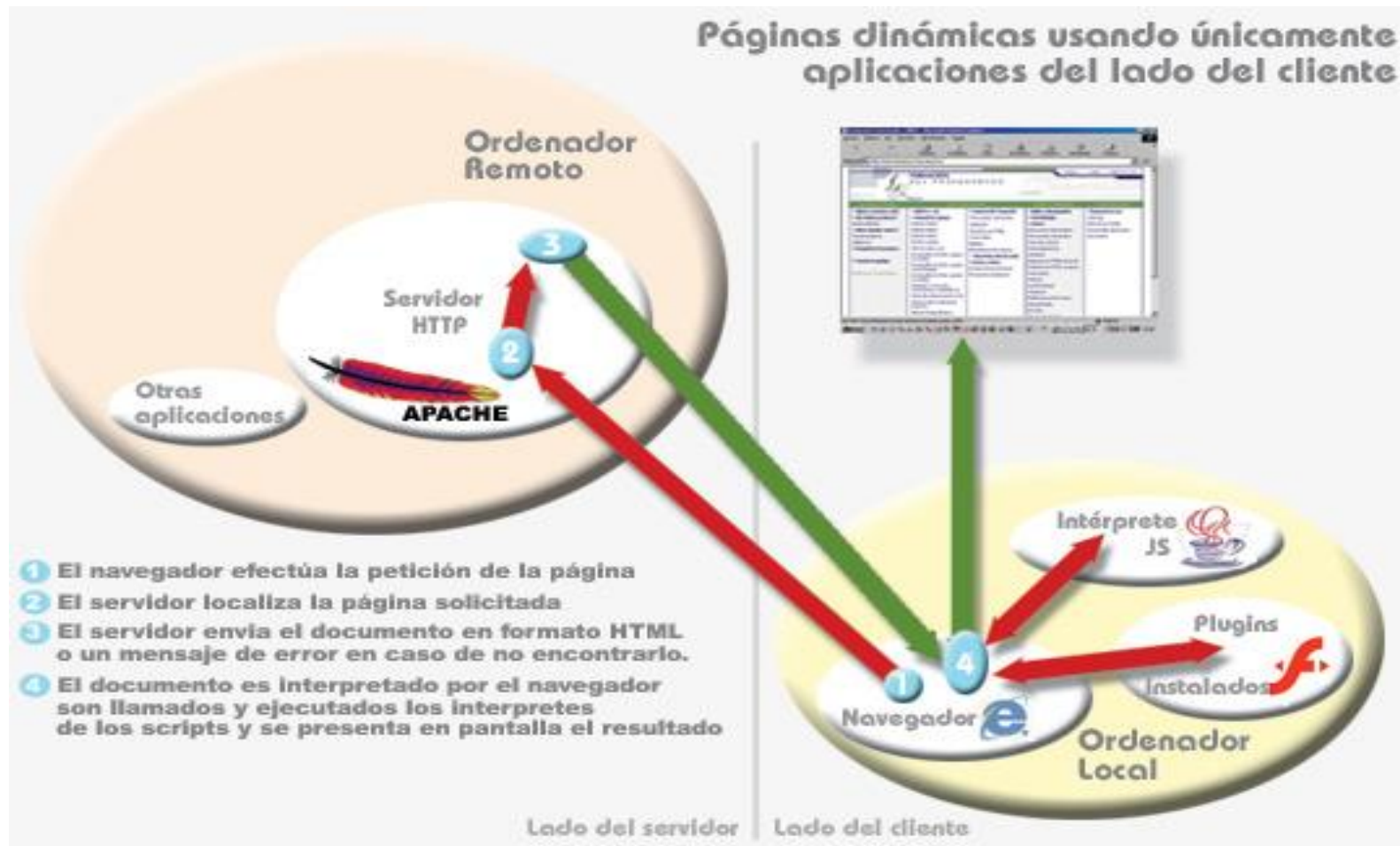
- **World Wide Web**

- Representa un universo de información accesible a través de Internet.
- Conjunto de recursos interconectados.
 - Componentes físicos: hubs, repetidores, puentes, routers, ...
 - Protocolos de comunicación: TCP, IP, HTTP, FTP, ...
 - Sistema de nombres de dominio (DNS)
 - Software para proveer y consumir dichos recursos: servidores y clientes.
- Configuración arquitectónica habitual basada en el modelo Cliente/Servidor.
 - CLIENTE: Componente consumidor de servicios
 - SERVIDOR: Proceso proveedor del servicio

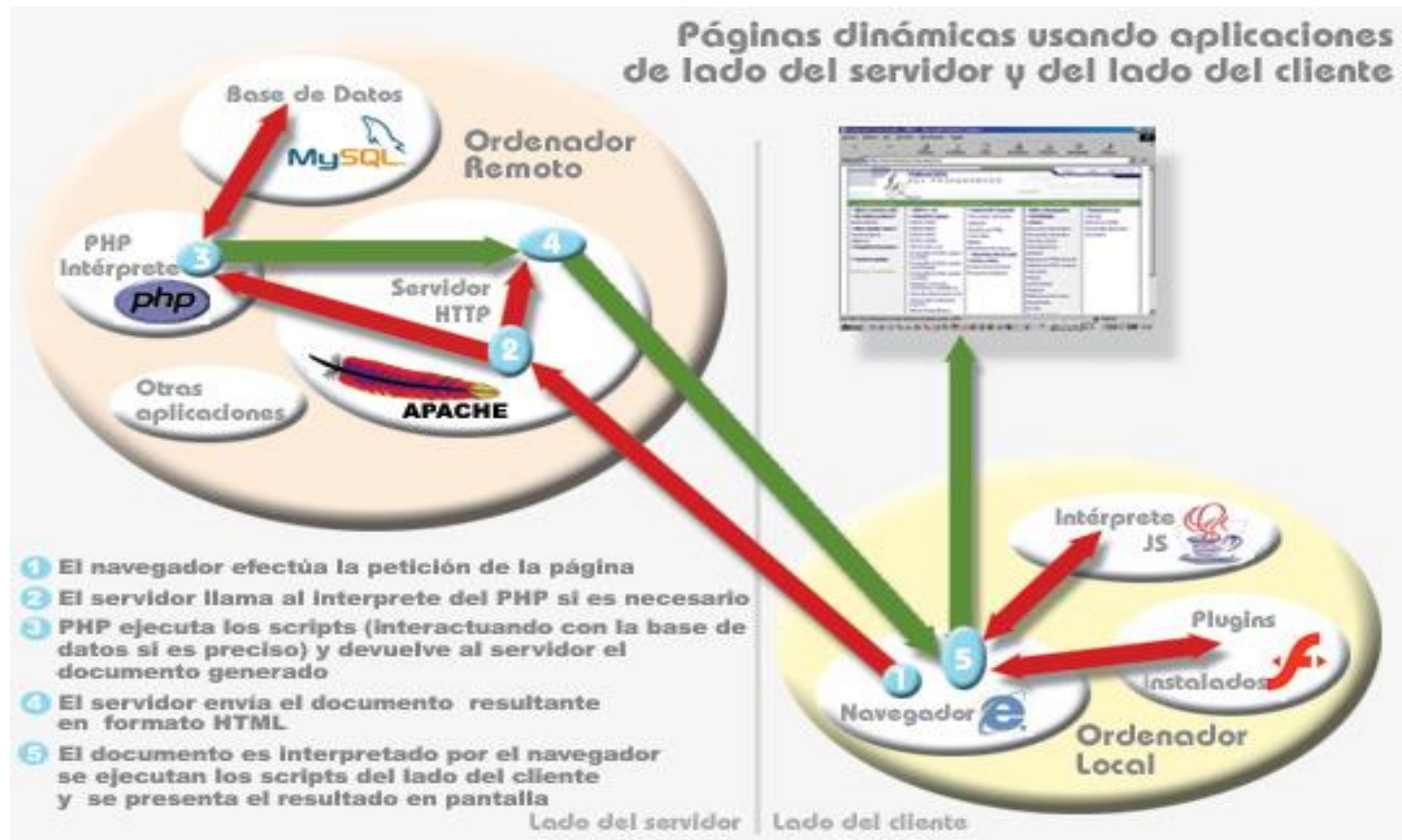
1. Evolución y características de los navegadores Web



1. Evolución y características de los navegadores Web



1. Evolución y características de los navegadores Web



1. Evolución y características de los navegadores Web

- **Navegador Web**
 - Software Cliente para acceder a contenidos de los servidores de Internet.
 - Diferentes tipos en función de sus capacidades, lenguajes soportados o facilidad de configuración.
 - La mayoría soporta DHTML (Dynamic HTML) para otorgar mayor funcionalidad.
 - “Un navegador es una aplicación distribuida habitualmente como software libre, que permite a un usuario acceder a un recurso publicado por un servidor web descrito mediante una dirección URL (Universal Resource Locator)”

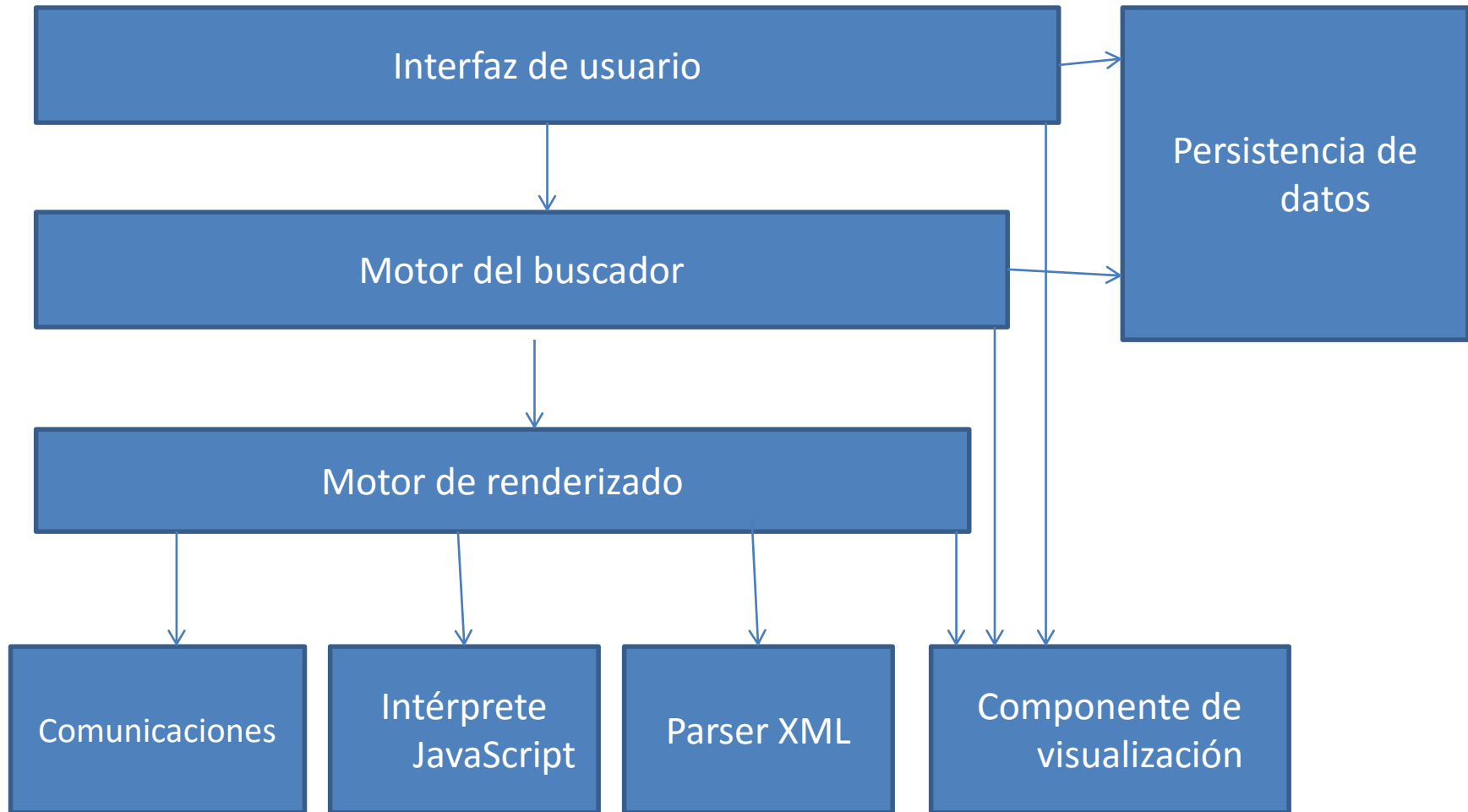
1. Evolución y características de los navegadores Web

- Criterios para diferenciar los navegadores:
 - **Plataforma de Ejecución.**
 - No todos los navegadores se pueden usar en cualquier ordenador.
 - Safari es exclusivo de Apple, pero tiene versiones para Windows.
 - **Características del navegador.**
 - La mayoría añaden funcionalidades asociadas a la experiencia del usuario: administración de marcadores, gestores de descarga, almacenamiento seguro de contraseñas y datos de formulario, corrección ortográfica o definición de herramientas de búsqueda.
 - **Personalización de la interfaz.**
 - Soporte para la navegación por pestañas, bloqueadores de ventanas emergentes, integración con visualizadores de formatos de ficheros (PDF), opciones de zoom o funciones avanzadas de búsqueda de texto.
 - **Soporte de tecnologías Web.**
 - Nivel de soporte de tecnologías CSS, Java, lenguajes de scripting del cliente, RSS o Atom, XHTML, etc.
 - **Licencia de software.**
 - De código libre, como Mozilla (licencia GNU GPL) y Google Chrome (licencia BSD).
 - Propietarios, como Internet Explorer o Safari.
 - Salvo raras excepciones (OmniWeb) todos son gratuitos.

2. Arquitectura de ejecución

- La interacción usuario-navegador es, básicamente, la siguiente:
 - A. El usuario indica la dirección del recurso al que quiere acceder.
 - B. El navegador visualiza el recurso en la pantalla del usuario.
- Cada navegador web tiene una forma de interpretar esta interacción y puede centrarse más en:
 - Ofrecer una respuesta rápida
 - Mostrar una respuesta más fiel al contenido del recurso
 - Priorizar los aspectos de seguridad en las comunicaciones
 - ...
- Cada navegador, por tanto, está formado por una serie de elementos y componentes que conforman su **arquitectura**, aunque todas parten de una ***arquitectura de referencia*** básica y común a todos los navegadores.

2. Arquitectura de ejecución



2. Arquitectura de ejecución

- ARQUITECTURA DE REFERENCIA. **Subsistema de Interfaz de Usuario.**
 - Capa que actúa de interfaz entre el usuario y el motor de buscador.
 - Visualiza barras de herramientas.
 - Visualiza el proceso de carga.
 - Gestiona las descargas de forma inteligente.
 - Plasma las preferencias de configuración de usuario o impresión.
 - Puede comunicarse con el Sistema Operativo, en algunos casos, para el manejo de sesiones de usuario o el almacenamiento de preferencias.

2. Arquitectura de ejecución

- ARQUITECTURA DE REFERENCIA. **Subsistema del Motor del Buscador.**
 - Capa que actúa de interfaz de alto nivel para el motor de renderizado.
 - También llamado Motor de Navegación.
 - Carga la dirección determinada (URL).
 - Soporta los mecanismos básicos de navegación (página anterior o siguiente, recarga de la página, ...)
 - Gestiona las alertas de JavaScript
 - Consulta y administra las preferencias de ejecución del motor de renderizado.

2. Arquitectura de ejecución

- ARQUITECTURA DE REFERENCIA. **Subsistema de Renderizado.**
 - Encargado de producir una representación visual del recurso obtenido.
 - Interpreta el código de la página Web.
 - En función de las tecnologías soportadas será capaz de mostrar documentos HTML o XML, hojas de estilo CSS, imágenes e incluso contenido embebido (audio/vídeo),
 - Establece las dimensiones exactas de cada elemento a mostrar y la posición de éstos.
 - Los motores de renderizado más conocidos son:
 - Gecko (Mozilla y Galeon)
 - Trident (Internet Explorer)
 - WebKit (Chrome, Safari y Epiphany)
 - Presto (Opera)
 - Tasman (Internet Explorer para Mac)

2. Arquitectura de ejecución

- ARQUITECTURA DE REFERENCIA. **Subsistema de Comunicaciones.**
 - Implementa los protocolos de transferencia de ficheros y documentos (HTTP, FTP, ...)
 - Identifica la codificación de los datos obtenidos en función de su tipo (texto, audio, vídeo, ...) codificado en estándar MIME (Multipurpose Internet Mail Extensions).
 - Puede almacenar una caché de elementos accedidos recientemente.

2. Arquitectura de ejecución

- ARQUITECTURA DE REFERENCIA. **Intérprete de JavaScript.**
 - Analiza y ejecuta el código JavaScript intercalado en HTML.
 - Puede ser configurado, e incluso deshabilitado desde el motor de navegación o el motor de renderizado.
 - Cada navegador tiene sus propios módulos de interpretación, por lo que es posible que existan subsistemas intérpretes de otros lenguajes, como applets de Java, Ajax o ActionScript.

2. Arquitectura de ejecución

- ARQUITECTURA DE REFERENCIA. **Parser XML.**
 - Permite cargar en memoria una representación en árbol (árbol DOM, Document Object Model) de la página.
 - El acceso a los diferentes elementos de una página por parte del navegador es mucho más rápido.

NOTA: Un “parser” es un analizador sintáctico

2. Arquitectura de ejecución

- ARQUITECTURA DE REFERENCIA. **Componente de Visualización.**
 - Ofrece funcionalidades relacionadas con la visualización de los contenidos de un documento HTML en una página web.
 - Ofrece primitivas de dibujo y posicionamiento en una ventana, un conjunto de componentes visuales predefinidos (widgets) y un conjunto de fuentes tipográficas.
 - Está relacionado con las librerías de visualización del Sistema Operativo.

2. Arquitectura de ejecución

- ARQUITECTURA DE REFERENCIA. **Subsistema de persistencia de datos.**
 - Funciona como almacén de diferentes tipos de datos para los principales subsistemas del navegador.
 - Suelen estar relacionados con el almacenamiento de historiales de navegación y mantenimiento de sesiones de usuario en disco.
 - Incluye las preferencias de configuración del navegador o la lista de marcadores.
 - A bajo nivel, este sistema administra también los certificados de seguridad y las cookies.

3. Lenguajes y Tecnologías de programación en entorno cliente

- Son aquellos que se ejecutan en el navegador Web.
- El lenguaje cliente principal es el HTML (lenguaje de marcado de hipertexto, HyperText Markup Language), y sus variaciones DHTML y XHTML.
- Con el fin de mejorar la interactividad con el usuario se incluyen los lenguajes de script JavaScript o VBScript.
- Otros lenguajes son ActionScript (para crear contenido Flash) o Ajax (como tecnología de extensión a JavaScript para comunicación asíncrona).
- Los applets de Java y las CSS son otras tecnologías del entorno cliente.

3. Lenguajes y Tecnologías de programación en entorno cliente

- **XML (eXtensible Markup Language)**
 - Es lenguaje (o metalenguaje) de etiquetado con unas reglas muy estrictas de codificación.
 - Se usa para el intercambio de una gran cantidad de datos.
 - Puede adjuntar un DTD (Definición de Tipo de Documento) para la validación de un archivo de datos (válido y bien formado).
 - Su objetivo principal es describir datos para su transferencia eficiente y no mostrarlos.
 - Los navegadores actuales suelen mostrar los archivos XML de forma esquemática.

3. Lenguajes y Tecnologías de programación en entorno cliente

- **HTML (HyperText Markup Language)**
 - Es una particularización del lenguaje SGML (Standard Generalized Markup Language), un sistema para la organización y etiquetado de documentos (ISO 1986).
 - Es el lenguaje de marcas de texto más usado en la Web. Creado en 1989 por Tim Berners Lee a partir del concepto de hipertexto y el lenguaje de marcas SGML.
 - No es un lenguaje de programación. Se basa en el uso de un sistema de etiquetas cerrado aplicado a un documento de texto.
 - No necesita ser compilado, sino interpretado por el navegador a medida que se avanza en el documento.
 - Con él se puede hacer: organizar texto y objetos, crear listas y tablas y, obviamente, permitir los hipervínculos (esencia de la Web).
 - Su evolución a dado lugar a lenguajes derivados que veremos a continuación.

3. Lenguajes y Tecnologías de programación en entorno cliente

- **XHTML (eXtensible HTML)**

- No es más que una adaptación de HTML al XML.
- Pretende un etiquetado más estricto que el HTML.

- **DHTML (Dynamic HTML)**

- Consiste en una forma de aportar interactividad a las páginas web.
- Permite la integración de HTML con lenguajes de scripting, hojas de estilo personalizadas y la identificación de los contenidos de una página web en formato de árbol (DOM).
- Permite aumentar la funcionalidad e interactividad de una página web: crear efectos, animaciones, juegos, ... y crear un auténtico entramado de capas.

3. Lenguajes y Tecnologías de programación en entorno cliente

- **CSS (Cascade Style Sheets)**

- Las hojas de estilo sirven para separar el formato que se quiere dar a la página de la estructura de ésta y demás instrucciones.

- **JavaScript**

- Lenguaje de programación de scripting (interpretado por el navegador) y embebido en un documento HTML.
- No es orientado a objetos, pero sí permite crear “pseudoclases” u objetos definidos por el usuario y, por supuesto, mantiene una jerarquía de objetos, tanto nativos como del lenguaje.
- Es débilmente tipado.
- Permite mejoras en la interfaz del usuario y la creación de páginas dinámicas.
- Tiene una sintaxis similar a C, aunque adopta nombres y convenciones propias de Java (aunque no tiene ninguna relación con este último lenguaje).
- Actualmente existen dos estándares alineados: ECMAScript e ISO/IEC 16262.
- Todos los navegadores modernos lo interpretan.

3. Lenguajes y Tecnologías de programación en entorno cliente

- **Applets de Java**

- Son pequeños componentes (objetos independientes) que se insertan en una página para incluir funcionalidades complejas.
- Son fragmentos de código Java que se ejecutan en el cliente y se benefician de la potencia y flexibilidad de este lenguaje.
- Los applets se programan en Java y se envían al cliente precompilados.
- Son, por tanto, menos dependientes del navegador que JavaScript e, incluso, independientes del Sistema Operativo.
- Son más lentos de procesar y no tienen acceso a ningún otro componente de la página.

3. Lenguajes y Tecnologías de programación en entorno cliente

- **Ajax (Asynchronous JavaScript And XML)**
 - Conjunto de técnicas y métodos de desarrollo web para la creación de aplicaciones web interactivas.
 - Con Ajax se mantiene una comunicación asíncrona con el servidor en segundo plano, al contrario que con HTML, que pierde la comunicación con el servidor cuando termina de cargarse la página.
 - Se puede, por tanto, realizar cambios sobre las páginas del cliente sin que se necesite recargarlas. Esto implica un aumento en la velocidad y en la interactividad.
 - El fundamento de Ajax se encuentra en el uso de un objeto específico de JavaScript denominado ***XMLHttpRequest*** y aceptado por la mayoría de los navegadores actuales.
 - Es una combinación, pues de 4 tecnologías existentes: XHTML/HTML y CSS, DOM, JavaScript y XML.

3. Lenguajes y Tecnologías de programación en entorno cliente

- **Adobe Flash y ActionScript**
 - Flash es una tecnología de animación que utiliza ActionScript como lenguaje principal.
 - Se usa para crear gráficos y animaciones.
 - Actualmente está en claro declive ante otras tecnologías y lenguajes como HTML 5.
 - Su uso ha permitido crear aplicaciones interactivas de gran complejidad y visualmente muy atractivas, permitiendo aumentar el grado de interactividad del usuario con la página web.
 - Al ser animaciones de índole vectorial el consumo del procesador (y de batería de dispositivos móviles) es más elevado. Además es software propietario.

4. Integración de código con las etiquetas HTML

- En el mismo documento HTML

```
<script type="text/javascript">  
    alert("Prueba de JavaScript");  
</script>
```

- En un archivo externo

```
alert("Prueba de JavaScript"); // archivo mensaje.js en la carpeta js  
y luego insertarlo  
<script type="text/javascript" src="js/mensaje.js"></script>
```

- En elementos HTML

```
<input type="button" value="Púlsame" onclick="alert('Prueba de  
JavaScript');" />
```