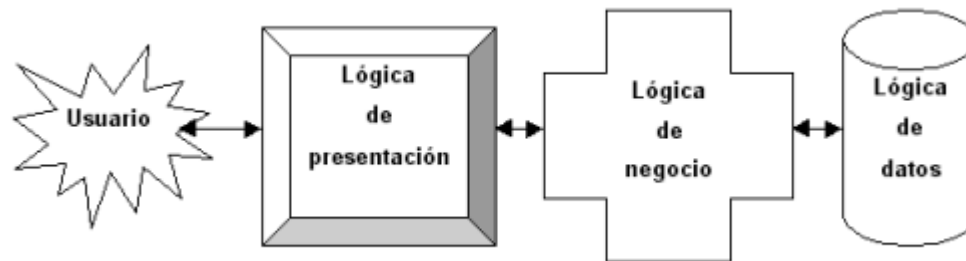


UD1. Arquitecturas y lenguajes de programación en clientes web

Desarrollo Web en Entorno
Cliente

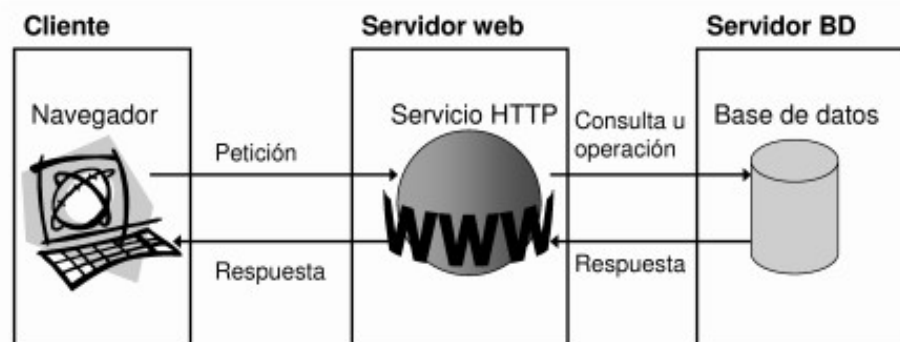
1. Arquitectura Cliente-Servidor

- Las aplicaciones web modernas siguen la arquitectura cliente-servidor. Normalmente:
 - Servidor: Ordenador potente dedicado a gestionar unidades de disco, tráfico de red, impresoras, datos o aplicaciones.
 - Cliente: Máquina menos potente que usan los recursos que ofrece el servidor.
- La arquitectura cliente-servidor permite la separación de funciones de una aplicación web:



1. Arquitectura cliente-servidor

- En una aplicación se distinguen 3 niveles:
 - Nivel superior: interacciona con el usuario. (cliente web).
 - Nivel intermedio: procesa los datos (servidor web).
 - Nivel inferior: proporciona los datos (BBDD).

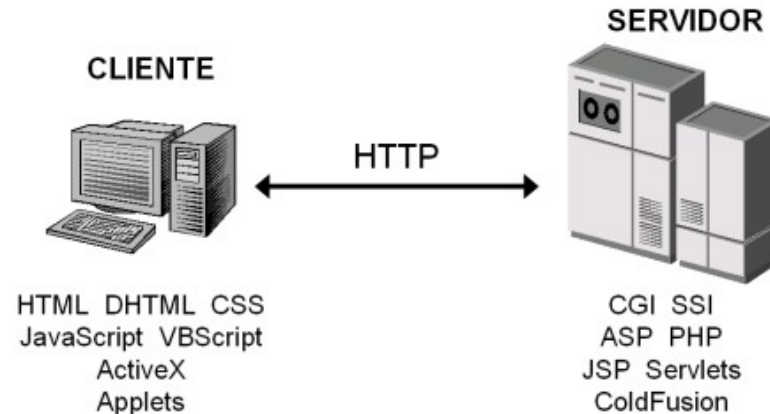


1. Arquitectura cliente-servidor

- Cliente y servidor se comunican mediante protocolo HTTP (nivel 7 modelo OSI).
- Aplicación web = parte cliente + parte servidor
 - Parte servidor:
 - Programas o scripts ejecutados en el servidor cuando el cliente solicita algunas páginas.
 - Recursos: imágenes, vídeos, etc.
 - Parte cliente:
 - Páginas HTML, scripts CSS, JavaScript, Applets de Java.

1. Arquitectura Cliente-Servidor

- Principales tecnologías:
 - Servidor: PHP, JSP, ASP,...
 - Cliente: HTML, CSS, JavaScript, Applets de Java,...



2. Tecnologías de programación en entorno cliente

- HTML: Lenguaje basado en etiquetas utilizado para la representación visual del contenido de una página web.
- CSS: Hojas de estilo utilizadas para dotar de formato a una página web.
- JavaScript: Lenguaje interpretado por el navegador empleado para dotar de interactividad a una página web. Hace uso de AJAX para comunicarse de forma asíncrona con el servidor sin necesidad de recargar la página.

2. Tecnologías de programación en entorno cliente

- Applets de Java: Fragmentos de código programados en Java que se ejecutan en el cliente. Se envían al cliente precompilados, y son ejecutados por una Máquina Virtual de Java.
 - Ventajas:
 - Potencia del lenguaje Java.
 - No dependen del navegador ni del s.o.
 - Inconvenientes:
 - Lentos de procesar.
 - No tiene acceso a todos los elementos de la página.

2. Tecnologías de programación en entorno cliente

- Adobe Flash y ActionScript:
 - Tecnología de animación bajo licencia de Adobe.
 - Utiliza ActionScript como lenguaje principal.
 - Permite la creación de gráficos y animaciones vectoriales.

3. Navegadores web

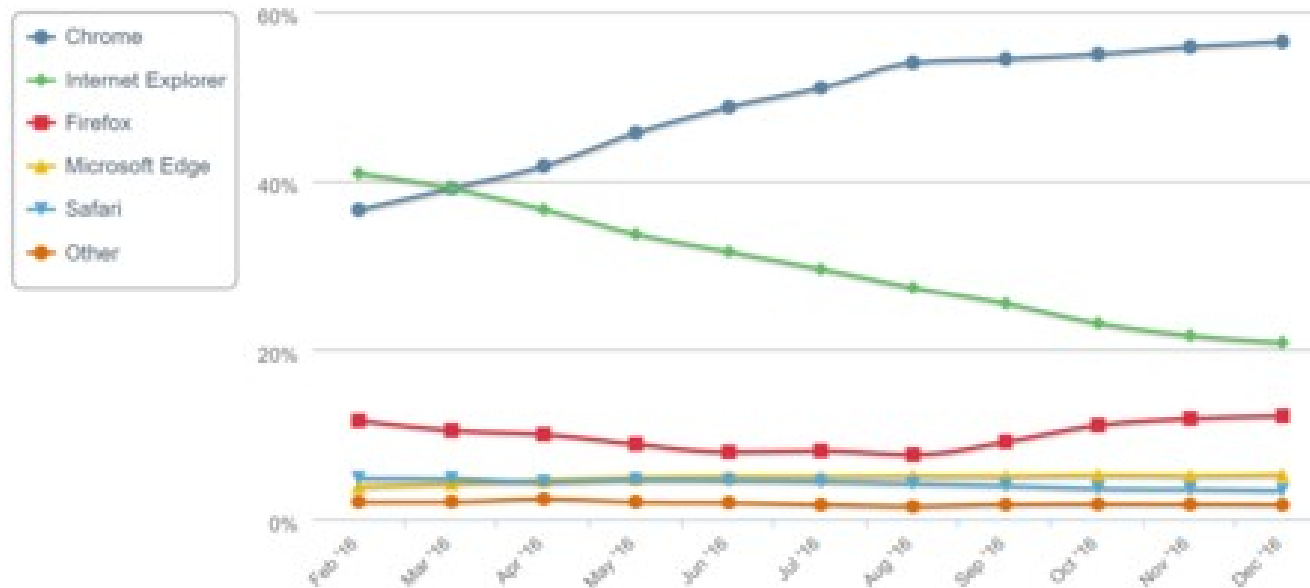
- Definiciones:
 - Componente software que se utiliza en el cliente y que permite acceder al contenido ofrecido por los servidores de Internet sin la necesidad de que el usuario instale un nuevo programa.
 - Aplicación que permite a un usuario acceder (y, normalmente, visualizar) a un recurso publicado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

3. Navegadores web

- Algunos navegadores importantes:
 - Mosaic: Primer navegador con capacidades gráficas.
 - Netscape: Primer navegador en incluir un módulo para la ejecución de código JavaScript.
 - Internet Explorer
 - Mozilla Firefox
 - Google Chrome
 - Safari

3. Navegadores web

- Estadística de uso de navegadores



3. Navegadores web

- Criterios de clasificación:
 - Plataforma de ejecución: Sistema operativo.
 - 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.
 - Tipo de licencia.

4. Entorno de desarrollo

- Visual Studio Code
 - Extensiones:
 - Auto Close Tag: Cierre de etiquetas.
 - Beautify: Formato a ficheros.
 - IntelliSense for CSS.: Autocompletado de propiedades CSS.
 - GitLens: Repositorios Git.
 - Open in browser: Ejecutar en navegador.
 - Spanish Language Pack

4. Entorno de desarrollo

- Instala y configura Visual Studio Code con las extensiones indicadas anteriormente. Crea un primer proyecto que conste de una página HTML estática y ejecútalo directamente en el navegador desde el propio editor.

5. Integración de JavaScript en documentos 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.

5. Integración de JavaScript en documentos HTML

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


5. Integración de JavaScript en documentos HTML

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

5. Integración de JavaScript en documentos HTML

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

5. Integración de JavaScript en documentos HTML

- JavaScript en un archivo externo:
 - Las mismas instrucciones de JavaScript que se incluyen entre bloques `<script></script>` pueden almacenarse en un fichero externo con extensión `.js`.
 - La forma de acceder y enlazar esos ficheros `.js` con el documento HTML 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.

5. Integración de JavaScript en documentos HTML

Archivo mensaje.js:

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

Archivo ejemplo2.html:

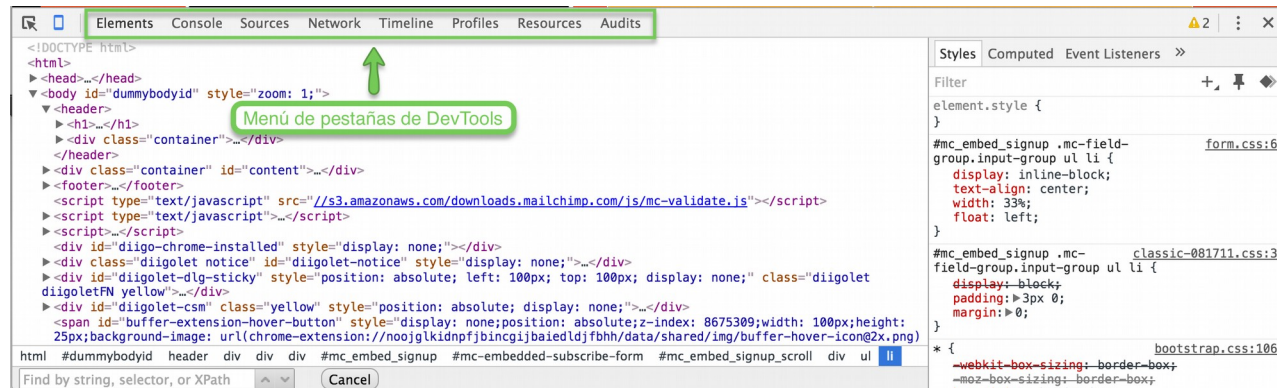
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=iso-8859-1" />
    <title>Ejemplo 2</title>
    <script type="text/javascript"
      src="/inc/mensaje.js"></script>
  </head>
  <body>
    <h1>Ejemplo 2: fichero externo</h1>
  </body>
</html>
```

6. Herramientas del navegador para desarrolladores.

- Extensiones del navegador que permiten, entre otras cosas:
 - Visualizar y modificar el código HTML.
 - Probar estilos y reglas CSS.
 - Comprobar las peticiones realizadas al servidor y su respuesta.
- Los más utilizados:
 - **Chrome Dev Tools** de Google Chrome
 - Firebug de Firefox.

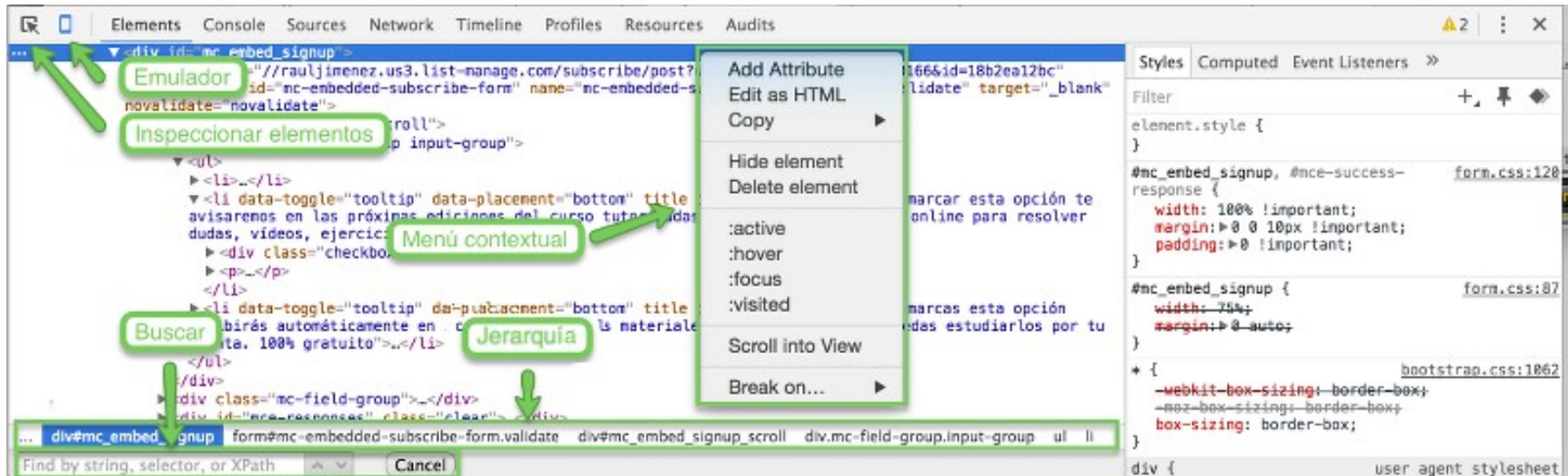
7. Chrome Dev Tools

- 3 grupos de herramientas principales:
 - Elementos (Elements): nos permite ver y modificar el código que representa la página que estamos viendo.
 - Fuentes (Sources): nos permite navegar por todos los ficheros (HTML, CSS y JavaScript) que utiliza la página que estamos viendo.
 - Red (Network): esta pestaña nos permite ver los recursos que recupera nuestro navegador usando peticiones HTTP mientras cargamos y usamos la página.



7. Chrome Dev Tools

- Pestaña Elements:
 - Representa el código HTML en forma de árbol.
 - Es posible editar el código, buscar texto o reordenar etiquetas.

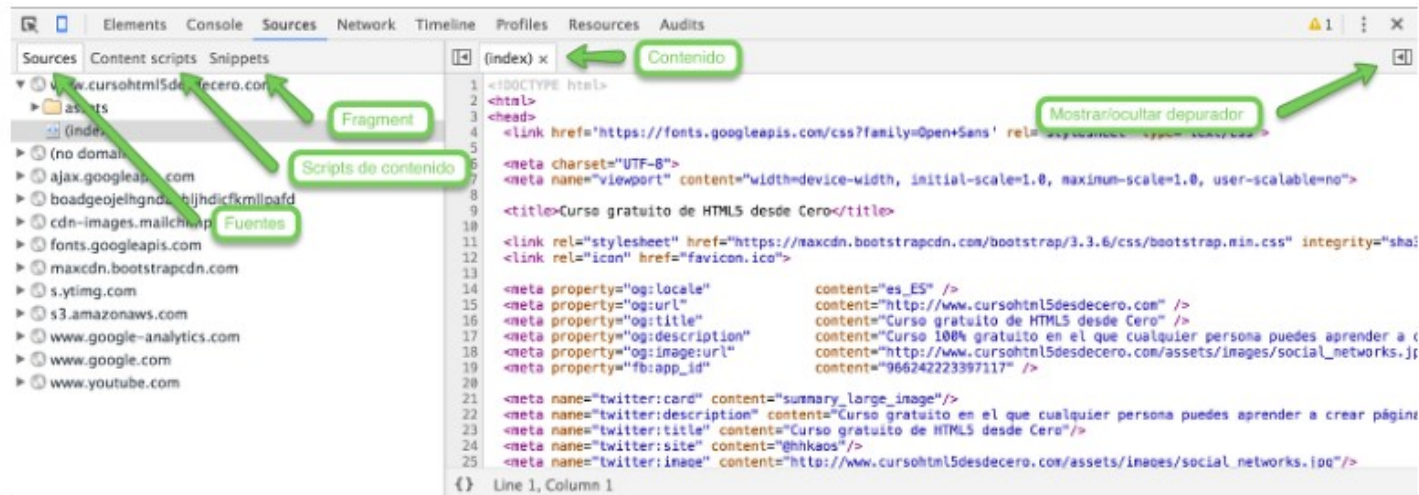


7. Chrome Dev Tools

- Pestaña Elements:
 - Emulador: esta opción nos permitirá simular que estamos usando un móvil o tablet.
 - Inspeccionar elementos: activando esta opción podrás hacer clic sobre cualquier parte de la página y el inspector señalará el código que representa el elemento seleccionado.
 - Menu Contextual:
 - Add attribute: permite añadir un atributo.
 - Edit as HTML: nos permite añadir, editar o quitar cualquier cosa.
 - Copy: permite copiar el elemento o el código y pegarlos.
 - Delete: permite eliminar el elemento.
 - Buscar: Nos permite buscar cualquier palabra dentro del código HTML.
 - Jerarquía: nos muestra todos los ancestros del elemento y nos permite seleccionarlos.

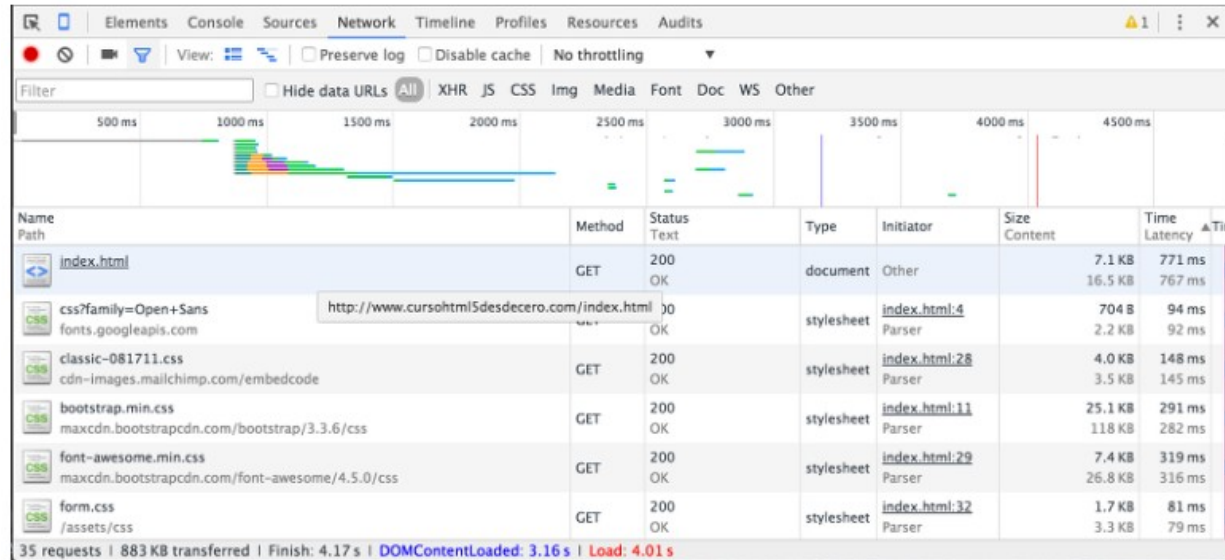
7. Chrome Dev Tools

- Pestaña Sources: Muestra las fuentes que se han utilizado para construir la página. Permite la modificación de ficheros HTML y CSS y depurar el código JavaScript.



7. Chrome Dev Tools

- Pestaña Network: Permite ver los recursos que solicita el navegador de un servidor usando peticiones HTTP, y los detalles de las mismas: tipo de mensaje (GET/POST), código de respuesta, etc.



7. Chrome Dev Tools

- Demo:
 - <https://googlechrome.github.io/devtools-samples/debug-js/get-started>