

## PARTE 1: ARQUITECTURAS , LENGUAJES Y TECNOLOGÍAS EN PROGRAMACIÓN WEB

Estudio sobre la arquitectura Cliente-Servidor, la arquitectura de referencia de los navegadores y su relación con los principales lenguajes y tecnologías de programación web

### Objetivos:

- Comprender la arquitectura Cliente-Servidor.
- Comprender la arquitectura de referencia de los navegadores.
- Relacionar los principales lenguajes y tecnologías de programación web con la arquitectura Cliente-Servidor y la arquitectura de referencia de los navegadores.

1. Rellena las celdas vacías de la siguiente tabla con las palabras “cliente” y/o “servidor” indicando si las acciones o archivos de la columna de la izquierda son interpretados en el cliente y/o en el servidor y si son generados por el cliente o el servidor.

| TECNOLOGÍA o LENGUAJE       | INTERPRETADO EN.. | GENERADO EN.. |
|-----------------------------|-------------------|---------------|
| Documento HTML              |                   |               |
| Documento CSS               |                   |               |
| Código JavaScript           |                   |               |
| Consulta MySQL              |                   |               |
| Código PHP                  |                   |               |
| Archivo FLASH               |                   |               |
| Applets de Java             |                   |               |
| Imagen pedida desde un HTML |                   |               |

2. Teniendo en cuenta los componentes de la arquitectura de referencia de los navegadores rellena las celdas vacías de la siguiente tabla con un “interpretarlo” o “recibirlo”, para indicar para cada tecnología o lenguaje qué componentes son los

encargados de interpretarlos o generarlos. Deja el espacio vacío si el componente no interfiere con la tecnología.

| TECNOLOGÍA o LENGUAJE       | SUBSISTEMA DE RENDERIZADO | SUBSISTEMA DE COMUNICACIONES | INTÉRPRETE DE JAVASCRIPT |
|-----------------------------|---------------------------|------------------------------|--------------------------|
| Documento HTML              |                           |                              |                          |
| Documento CSS               |                           |                              |                          |
| Código JavaScript           |                           |                              |                          |
| Consulta MySQL              |                           |                              |                          |
| Código PHP                  |                           |                              |                          |
| Archivo FLASH               |                           |                              |                          |
| Applets de Java             |                           |                              |                          |
| Imagen pedida desde un HTML |                           |                              |                          |

3. Es posible que el código PHP pueda ejecutarse en el Navegador una vez hemos recibido la página HTML? Porqué?
  
4. Visita [http://sheldonbrown.com/web\\_sample1.html](http://sheldonbrown.com/web_sample1.html), y utiliza la consola del navegador para determinar cuantas peticiones ha de realizar el navegador para mostrar correctamente la web. Indica también que recursos pide en cada petición
  
5. En una arquitectura Cliente-Servidor, indica como mínimo cuantas peticiones ha de realizar el navegador (explicando el motivo de cada consulta) para mostrar correctamente un archivo HTML con **4 imágenes**, **3 documentos CSS** vinculados, **2 documentos JavaScript** vinculados y **1 archivo Flash**.

## PARTE 2: INTRODUCCIÓN AL DESARROLLO CON JAVASCRIPT

**Introducción a la sintaxis de JavaScript y a algunas de sus herramientas de desarrollo y depuración.**

**Objetivos:**

- Comprender las principales características de los lenguajes de Script ejecutados en un Navegador.
- Aprender la sintaxis básica de la programación en JavaScript
- Utilizar herramientas de generación y depuración de código.

6. En una nueva carpeta de nombre T1 crea un nuevo documento HTML con el nombre “index.html”, añade en el *body* el siguiente código, visualiza la página web en el navegador y adjunta una captura de pantalla con el resultado.

```
<header>
  <h1> Introducción a JavaScript </h1>
</header>
<article>
  <script type="text/javascript">
    document.write("<b>Document.write</b> nos permite escribir texto o tags HTML");
    document.write("<br />");
    document.write("pero solo mientras se carga la web. ");
    document.write("Si lo utilizamos una vez cargada, se borrará todo el contenido HTML");
  </script>
  <noscript>Si se muestra este mensaje, tu navegador no soporta JavaScript. </noscript>
</article>
```

7. Crea un nuevo documento JavaScript con el nombre “ejercicioIO.js”, vincúlalo a “index.html” y añade en “ejercicioIO.js” una función “prompt()” con el mensaje: “Cuando se ejecuta el JavaScript?” que recoja la respuesta en una variable.
8. Añade al JavaScript “ejercicioIO.js” una función “alert()” que muestre el valor introducido en el “prompt()”.
9. Crea y vincula con el HTML un nuevo documento JavaScript con el nombre “ejercicioSupInf.js” que pida dos números al usuario mediante dos “prompt()” y muestre un “alert()” indicando cual de los dos es superior y cual inferior a no ser que

los dos sean iguales, en tal caso ha de indicar que los dos son iguales. En todos los casos se han de mostrar los números introducidos.

10. Crea y vincula con el HTML un nuevo documento JavaScript con el nombre "ejercicioDia.js" que pida por "prompt()" el nombre del día de la semana (pej: "lunes", "martes", etc..) y utilizando una estructura de control "switch" almacene en una variable el número del día correspondiente empezando por lunes y una vez finalizado el "switch" muestre un "alert()" indicando el número de la semana correspondiente. pej: si se introduce "lunes", se debe mostrar el mensaje "día de la semana 1", si se introduce "domingo", se debe mostrar el mensaje "día de la semana 7".
  
11. Crea y vincula con el HTML un nuevo documento JavaScript con el nombre "numeroMayor.js" que pida números al usuario (por "prompt()") mientras el usuario no introduzca un número superior a 50. Si el usuario introduce un número superior o igual a 50 ha de mostrar un "alert()" con un mensaje que indique cuantas veces se ha introducido un número menor a 50.