

1.¿Qué es la Web?

Sistema de distribución de información basada en hipertexto o hipermedios enlazados y accesibles a través de Internet, que fue creada por Tim Berners-Lee en 1989.

2. Haz un esquema de la interacción entre un servidor web y el navegador en el caso de solicitud de contenidos dinámicos.

1. Introduce URL (Navegador web).
2. Busca la IP (Internet).
3. Solicita la página principal (Navegador web).
4. Recibe la petición (Servidor web).
5. Obtiene página del disco duro (Disco).
6. Contiene PHP (Servidor web).
7. Procesa PHP (Procesador PHP).
8. Ejecuta SQL(MySQL).
9. Recibe los datos (Procesador PHP).
10. Devuelve la página(Servidor Web).
11. Muestra la página(Navegador web).

3. Haz un esquema de la interacción entre un servidor web y el navegador en el caso de solicitud de contenidos estáticos.

1. Introduce URL.(Navegador web)
2. Busca la IP. (Internet)
3. Solicita la página principal (Navegador web).
4. Recibe la petición (Servidor web).
5. Obtiene página del disco duro (Disco).
6. Devuelve la página(Servidor Web).
7. Muestra la página(Navegador web).

4. Explica las principales diferencias entre aplicaciones web “clásicas” y aplicaciones AJAX.

AJAX(Asynchronus JavaScript and XML) es una técnica de desarrollo web que permite crear aplicaciones interactivas y gestionar una comunicación asíncrona con el servidor.

Las principales diferencias son:

1- Las aplicaciones web clásicas mandan y reciben peticiones HTTP mientras que en AJAX el navegador envía una petición HTTP y recibe una respuesta en XML que AJAX transforma en HTML para la interfaz de usuario.

2-En las aplicaciones web clásicas el modelo de aplicación es síncrono mientras que en AJAX es asíncrono.

3- En las aplicaciones clásicas el usuario va de una página a otra mediante la recarga completa de páginas. En AJAX el flujo de datos se define mediante software en el cliente y la comunicación con el servidor realizando cambios concretos en la página. Debido a esto las peticiones no se conservan en el historial mientras que en las aplicaciones clásicas si.

4- Respecto al tráfico en la red. AJAX tiene una carga inicial elevada y después es muy reducida. En las aplicaciones clásicas comienzan con poca cantidad de datos y va aumentando con el tiempo.

5- Es difícil indexar páginas con AJAX en los motores de búsqueda. Esto no ocurre con las aplicaciones clásicas.

5. Acceso a MySQL con PHP. Escribe el esquema de acceso y pon código de ejemplo.

1. Conexión
`$db_cons=mysql_connect('localhost','usuario','clave');`
2. Seleccionar la BD
`mysql_select_db('mi_base_de_datos');`
3. Construir la consulta.
`$consulta='select * from tabla_alumno';`
4. Ejecutar la consulta.
`$result= mysql_query($consulta);`
5. Recuperar los datos y construir la página web:
`$num_filas=mysql_num_rows($result);
for($i=0;$i<=$num_filas;$i++){
 $fila=mysql_fetch_row($result);
}`
6. Desconectar
`mysql_close('mi_base_de_datos');`

6. Explica que es una cookie. Pon un ejemplo de creación y eliminación.

Mantienen información sobre los usuarios y se intercambian durante la transferencia de las cabeceras antes del envío de la página HTML.

Para crear una cookie utilizamos setcookie de la siguiente forma:

`setcookie(name, value, expire, path, domain, secure, httponly);`

Un ejemplo de creación de cookie sería:

`setcookie('nombre','valor',time()+60*60*24*7,'/');`

Para eliminar una cookie tenemos que utilizar los mismos datos que en la creación pero sin contenido:

```
setcookie('nombre','',-1,'');
```

7. Explica en qué consiste el Digital Object Model (DOM). Pon un ejemplo ilustrando su uso.

Es una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML; es un modelo de cómo pueden combinarse dichos objetos y una interfaz para manipularlos.

A través de DOM los programas pueden acceder y modificar el contenido de los documentos HTML y XML.

JavaScript está diseñado en torno al Modelo de Objetos Documentales (DOM): Las partes de un documento HTML son objetos, cada uno con sus variables de instancia y métodos.

Ejemplo:

```
<script>
    url=document.links.linkname.href;
    document.write('la URL es '+url);
</script>
```

8. Diferencias y ventajas entre desarrollo de aplicaciones web en el servidor y en el cliente.

Las aplicaciones que se ejecutan en el lado del cliente se ejecutan con poca carga computacional. Permite interactividad y ser más rápidas (efectos de animación, gráficos, validaciones). En cambio supone un tiempo de carga inicial al cargarse la página.

Las aplicaciones en el lado del servidor permiten devolver contenido dinámico al usuario que ha realizado una petición. Pueden ejecutar consultas a bases de datos. Un inconveniente es que se tiene que realizar peticiones constantemente al servidor.

9. Estructura de los mensajes en el protocolo HTTP.

Las peticiones y respuestas tienen la siguiente estructura:

1. Línea inicial (petición-respuesta)
2. Línea de cabecera (metadatos)
3. Cuerpo del mensaje. Datos que se intercambian.

10. Describe cómo se manejan archivos con PHP. Pon ejemplos.

- Podemos abrir archivos en local: `file_exists('archivo.txt');`
- Manejo de ficheros con las siguientes funciones:
 - + lectura-> `fopen('archivo.txt','r');` // Mirar el parentesis que no se si es un 1
 - + escribir-> `fwrite($fh,$text);`
 - +cerrar-> `fclose($fh);`

11. Enumera los aspectos más importantes en el desarrollo de aplicaciones web.

- Interfaz web: funcionalidades para ofrecer una experiencia interactiva(AJAX)
- Consideraciones técnicas: en que clientes o SO se va a ejecutar nuestra aplicación
- Estructura: estructurar en 3 capas de aplicación.
 - Primera capa: el navegador web
 - Segunda capa: tecnologia de web dinámica(PHP, Javascript)
 - Tercera capa: base de datos
- Estrategia empresarial: un modelo de negocio que llame la atención de los clientes

12. Principales capas de servicio de la plataforma Cloud Computing

Está formado por 3 capas:

- SaaS(usuario final): es la capa más alta, se caracteriza por ofrecer una aplicación como un servicio, bajo demanda y se paga por el uso(no por licencia) Ej: dropbox, google apps.
- PaaS(desarrollador de aplicaciones): es la capa intermedia, contiene escritorio software con herramientas de desarrollo para cada plataforma. Ej: google app engine
- IaaS(arquitecto de red): es la capa final, permite el almacenamiento básico, capacidades de conjunto, servidores, sistemas de almacenamiento, etc. Engloba todo el hardware. Ej: amazon web services.

13. Definición de servicio web y enumera herramientas específicas para su desarrollo.

Son componentes de aplicaciones que se comunican a través de protocolos abiertos. Son autocontenidos y autodescriptivos; descubribles a través de UDDI y que son utilizables por otras aplicaciones. Se manejan a través de mensajes XML.

Herramientas (Estándares):

- XML-RPC
- WSDL
- SOAP
- UDDI

14. Explique la interacción entre las plataformas de Cloud Computing y las tecnologías web

Las tecnologías web sirven para acceder a los recursos de conocimiento disponibles en internet a través del navegador. El cloud Computing presta servicios a dichos recursos, de manera que los usuarios acceden a los recursos que proporcionan las tecnologías web y son ellos los que simplifican el uso de los recursos.

15.-CLOUD COMPUTING

Es un modelo de prestación de servicios de negocio y tecnologías. el usuario accede a estos servicios bajo demanda y paga únicamente por el uso realizado. Se caracteriza por ser muy flexible y adaptable. El usuario tiene la sensación de tener recursos ilimitados.

Características:

- Agilidad
- Costo
- Elasticidad y escalabilidad
- Independencia entre dispositivos y ubicación
- rendimiento
- Mantenimiento

Ventajas:

- Reducción de costes(infraestructura, ordenadores de usuario, licencias soft., energía, personal)
- Gestión(Más simple, actualizaciones instantáneas)
- Prestaciones (Disponibilidad permanente, capacidad ilimitada, adaptación elástica)
- Universalidad

16.-BIG DATA

Es un concepto que hace referencia al almacenamiento de grandes cantidades de datos y a los procedimientos usados para encontrar patrones repetitivos dentro de esos datos.

Las dificultades más habituales son la recolección, búsqueda, compartición y visualización.

La tendencia de manipular tantos datos provoca la necesidad de la creación de informes estadísticos y predictores.

Herramientas para tratar: Hadoop, noSQL, Cassandra, ...

17. Indica las principales características del lenguaje de Javascript explicando cómo ayudan en el desarrollo de aplicaciones web.

Lenguaje de alto nivel, débilmente tipado, incrustable en HTML, con soporte para PDO, interpretado y ejecutado en el navegador. Se puede utilizar con eventos asociados, en modo incrustado con las etiquetas script y en ficheros externos.

JavaScript está diseñado entorno al Modelo de Objetos Documentales (DOM): Las partes de un documento HTML son objetos, cada uno con sus variables de instancia y métodos.

Esto ayuda a que las aplicaciones web tengan menos carga computacional, más interactividad y sean más rápidas

18. ¿Cuál es el proceso que ocurre cuando se atiende un servicio de páginas web?

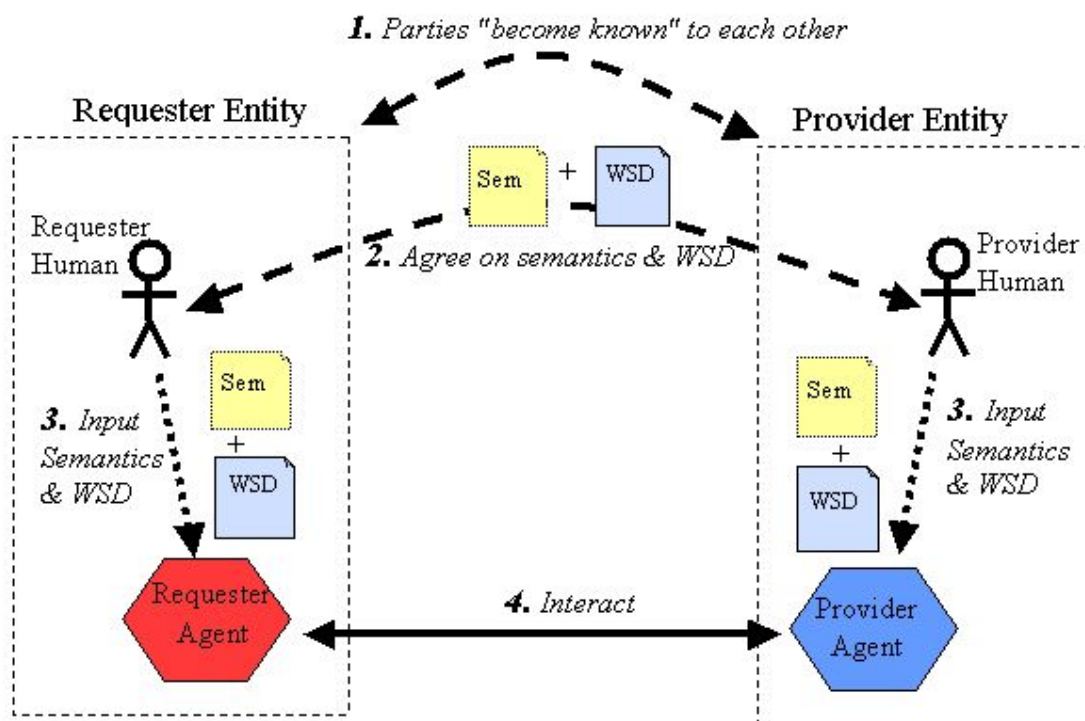


Figure 1-1. The General Process of Engaging a Web Service

- 1-Las partes se indentifican entre sí.
- 2-Llegan a un acuerdo en la semántica y en el WSD.
- 3- El usuario que pide el servicio introduce una semántica y el WSD.
- 4-El proveedor del servicio hace lo mismo.
- 5- Se produce la interacción entre ambas partes.