



Curso de XHTML

Unidad Didáctica 01: ¿Qué es HTML?



Ayuntamiento
de Vitoria-Gasteiz
Vitoria-Gasteizko
Udala

Índice de contenidos

- Introducción
- Glosario de términos
- Elementos de una arquitectura web
- Proceso de una una petición web
- Resumen de una petición sólo web
- Resumen de una respuesta sólo web
- Conclusiones



Introducción

A fin de entender los mecanismos que tienen que ver con el html es necesario conocer las bases del protocolo que lo pone en marcha

HTTP

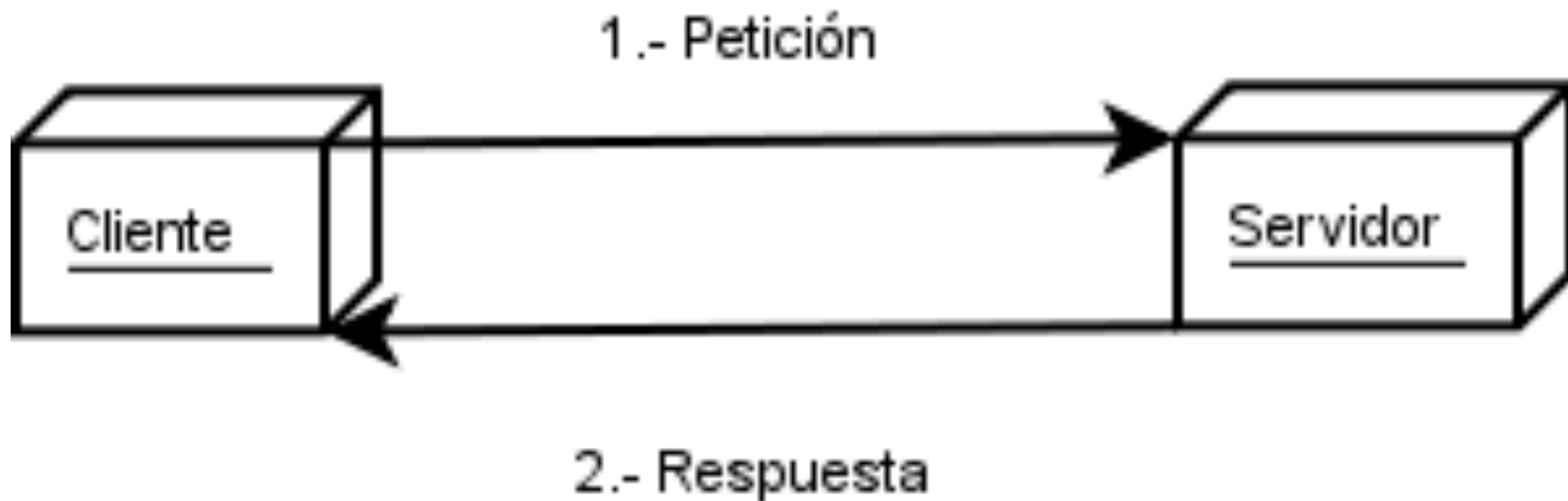


Introducción

- La arquitectura HTTP es importante describir el funcionamiento de la arquitectura de aplicaciones Cliente-Servidor.
- En Dicha arquitectura disponemos de dos tipos de agentes:
 - Cliente: Ordenador o programa informático que solicita información.
 - Servidor: Ordenador o programa informático que dispone de información.



Introducción



Introducción

- Dentro de este tipo de arquitectura podemos definir distintos tipos de Agentes dependiendo del tratamiento que se realice con la información:
 - Activo: El Agente realiza un procesamiento activo de la información. Cálculo intensivo, cálculo de la presentación gráfica de la información, acceso a datos son algunas de las tareas típicas de un Agente Activo.
 - Pasivo: El agente se limita a manejar información en bruto o preprocesada.



Introducción

- Tipos de Arquitectura Cliente Servidor
 - Cliente Activo, Servidor Pasivo: El cliente realiza la practica totalidad del trabajo de procesamiento de la información. Ejemplo: Google Earth.
 - Cliente Pasivo, Servidor Pasivo: Tanto el cliente como el Servidor simplemente pasan información. Ejemplo: Gateways de comunicaciones VoIP.
 - Cliente Pasivo, Servidor Activo: El Servidor realiza todo el trabajo de procesamiento y el cliente simplemente presenta los datos. Ejemplo: Servidores de terminales.
 - Cliente Activo, Servidor Activo: Tanto el Servidor como el Cliente procesan la información. Ejemplo: Servicios de Correo Electrónico.



Introducción

Dependiendo de la aplicación HTTP que el tipo de arquitectura puede variar, pero por simplicidad en el presente manual sólo va a tratarse del Tipo Cliente Activo - Servidor Activo.



Glosario de términos

- HTML: Hyper Text Markup Language. Lenguaje que permite definir el contenido y su presentación para que el cliente web permita visualizar la información que contiene. Dicho lenguaje permite la inclusión de medios (gráficos, sonidos, vídeos, etc.).
- Etiqueta o TAG: Unidad mínima de representación en el Lenguaje HTML. Permite definir: párrafos, tablas, enlaces, imágenes, etc...
- CSS: (Cascading Style Sheets) Hojas de Estilo en Cascada . Permiten generar clases de estilo que se pueden aplicar a etiquetas HTML. Dichas hojas de estilo se pueden definir dentro de los ficheros HTML o incluso referenciarlos. Los TAG permiten agrupar cambios en las propiedades de un determinando TAG.
- Java Script (JS): Lenguaje de programación que permite en la parte cliente procesar y realizar cambios en la presentación. Dichos Scripts suelen estar referenciados desde el fichero HTML.



Glosario de términos

- Página Web: Unidad de contenido de un servidor web. Normalmente está compuesta de un fichero formateado en HTML y sus referencias.
- Sitio Web: Conjunto de reglas del Servidor Web y las páginas Web que lo componen.
- Servicio DNS: Servidor que entre otras funciones permite traducir nombres de máquinas en Internet a direcciones IP.
- Navegador Web: Cliente que permite consultar a Servidores Web y presentar la información HTML. Ejemplos: Firefox, Konqueror, Opera, Internet Explorer, etc.
- Plugin Web: Extensión de un navegador web que permite añadir una funcionalidad a un navegador web, por ejemplo la visualización de elementos no estándar. Ejemplos: Flash Player, JVM, etc.



Glosario de términos

- Apache: Servidor Web Libre. Dicho servidor es altamente flexible y actualmente el 60% de los sitios web funcionan con Apache.
- Módulo de Apache: Conjunto de funcionalidades agrupadas que permiten extender el tratamiento de la información por parte del servidor Web Apache.
- URL: Uniform Resource Locator. Definición del acceso a un recurso. El concepto de URL ha sido incorporado dentro del más general de URI (Uniform Resource Identifier - Identificador Uniforme de Recurso), pero el término URL aún se utiliza ampliamente. La URL está dividida en tres partes: Nombre del Servidor Web, Recurso a acceder y Parámetros GET.
- Método GET: Forma de paso de parámetros mediante URL.
- Método POST: Forma de paso de parámetros mediante formulario (normalmente).
- PHP: PHP Hypertext Pre-processor. es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP está preparado para funcionar como un módulo de Apache (y otros servidores Web).



Glosario de términos

- Dirección IP: Identificativo único de un interfaz de red basado en el Protocolo IP.
- SGBD: Sistema de Gestión de Bases de Datos. Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.
- SQL: Lenguaje de Consulta Estructurado (Structured Query Language). Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Auna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla.

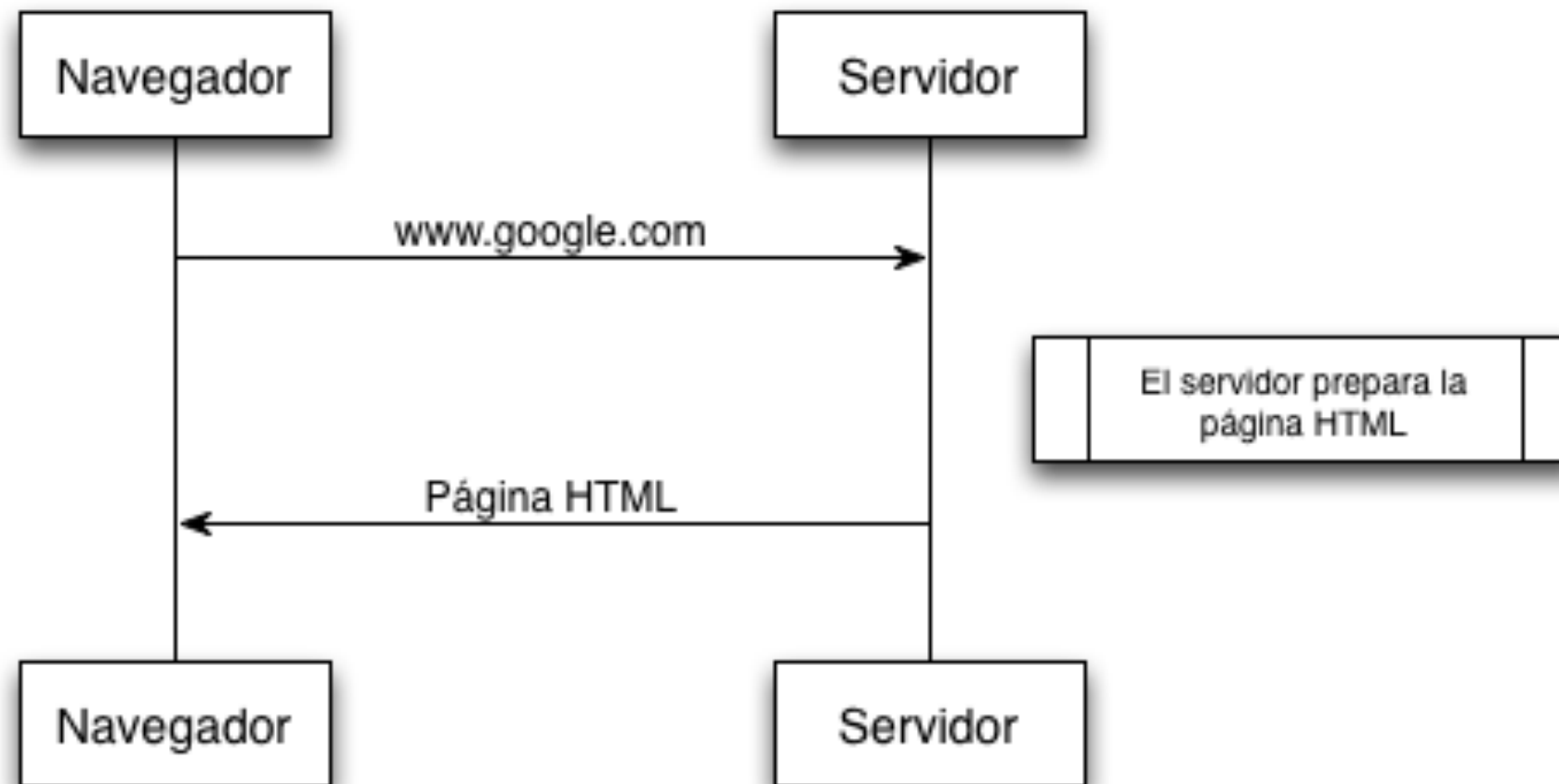


Elementos de una Arquitectura web

- Cliente Web/DNS: Navegador Web. Por ejemplo: Firefox.
- Servidor DNS: Conversor de nombres a IP's. Por ejemplo: BIND.
- Servidor Web: Almacén y procesador de la Información Web. Apache en nuestro caso.
- Módulo de PHP: Módulo para apache que permite procesar ficheros escritos en el Lenguaje PHP. Por ejemplo: PHP5
- Aplicación LAMP: Conjunto de ficheros PHP que componen una Aplicación Web. Por ejemplo: PhpMyAdmin
- SGBD: Sistema de gestión de bases de datos que permitirá almacenar la información dinámicamente. En nuestro caso Mysql.



Resumen una petición web



Resumen de una petición web

Cada conversación en la web comienza con una petición. Esta petición es un mensaje de texto creado por un cliente (navegador, app) en un formato especial conocido como HTTP. El cliente envía esta petición a un servidor, y entonces espera la respuesta.



Resumen de una petición web

- Una petición en lenguaje HTTP sería algo así:

GET / HTTP/1.1

Host: google.com

Accept: text/html

User-Agent: Chrome/31.0.1650.57 (Macintosh; Intel Mac OS X 10_9_0)

<http://cursosdedesarrollo.com/>



Resumen de una petición web

- Este simple mensaje comunica todo lo necesario acerca de qué recurso exactamente está solicitando el cliente.
- La primera línea de una petición HTTP es la más importante y contiene dos cosas:
 - el URI (Uniform Resource Identifier)
 - el método HTTP
- El URI (e.g. /, /contact, etc) es la única dirección o ubicación que identifica el recurso que el cliente quiere.



Resumen de una petición web

- El método HTTP (e.g. GET) define lo que quieres hacer con el recurso. Los métodos HTTP son los verbos de la petición y definen las pocas maneras comunes que pueden actuar sobre el recurso:
 - GET Recuperar el recurso del servidor
 - POST Crear un recurso en el servidor
 - PUT Actualizar el recurso en el servidor
 - DELETE Borrar el recurso del servidor



Resumen de una petición web

- Teniendo esto en cuenta, es muy fácil imaginarse cómo debería ser una petición HTTP para eliminar una entrada específica de un blog, por ejemplo:
 - DELETE /blog/15 HTTP/1.1
- Aunque hay nueve métodos HTTP definidos por la especificación HTTP, muchos de ellos no son muy utilizados. En realidad, muchos navegadores modernos no son compatibles con los métodos PUT y DELETE.



Resumen de una petición web

- Además de la primera línea, una petición HTTP contiene invariablemente otras líneas de datos llamadas request headers o cabeceras. Estas cabeceras pueden suministrar una amplia gama de información como el servidor (Host), los formatos de respuesta que acepta el cliente (Accept) y la aplicación que utiliza el cliente para realizar la solicitud (User-Agent).



Resumen de una respuesta web

Una vez que el servidor ha recibido la petición, sabe exactamente qué recurso necesita el cliente (via URI) y qué es lo que el cliente quiere hacer con ese recurso (via método HTTP).



Resumen de una respuesta web

- Traducido a HTTP, la respuesta enviada al navegador será algo como esto:

HTTP/1.1 200 OK

Date: Sun, 01 Dec 2013 18:17:45 GMT

Server: Apache/2.2.22 (Ubuntu)

Content-Type: text/html

<html>

<!-- ... -->

</html>

<http://cursosdedesarrollo.com/>



Resumen de una respuesta web

- La respuesta HTTP contiene el recurso solicitado (el contenido HTML en este caso), así como otra información acerca de la respuesta. La primera línea es especialmente importante y contiene el código de estado de respuesta HTTP (200 en este caso). El código de estado comunica el resultado global de la solicitud. Existen diferentes códigos de estado que nos indican éxito, error o que el cliente necesita hacer algo (e.g. redirigir a otra página)



Resumen de una respuesta web

- Los códigos de estado más comunes son:
 - 200 OK. Indica éxito.
 - 304 Not Modified. Esto muestra que el recurso en cuestión no ha cambiado y que el navegador debe cargarlo desde su caché.
 - 404 Not Found. Esto sugiere que el recurso no se encuentra en el servidor.
 - 401 Authorization Required. Esto indica que el recurso está protegido y quiere unos credenciales válidos antes de que el servidor pueda conceder el acceso.
 - 500 Internal Error. Esto significa que el servidor ha tenido un problema procesando la petición.

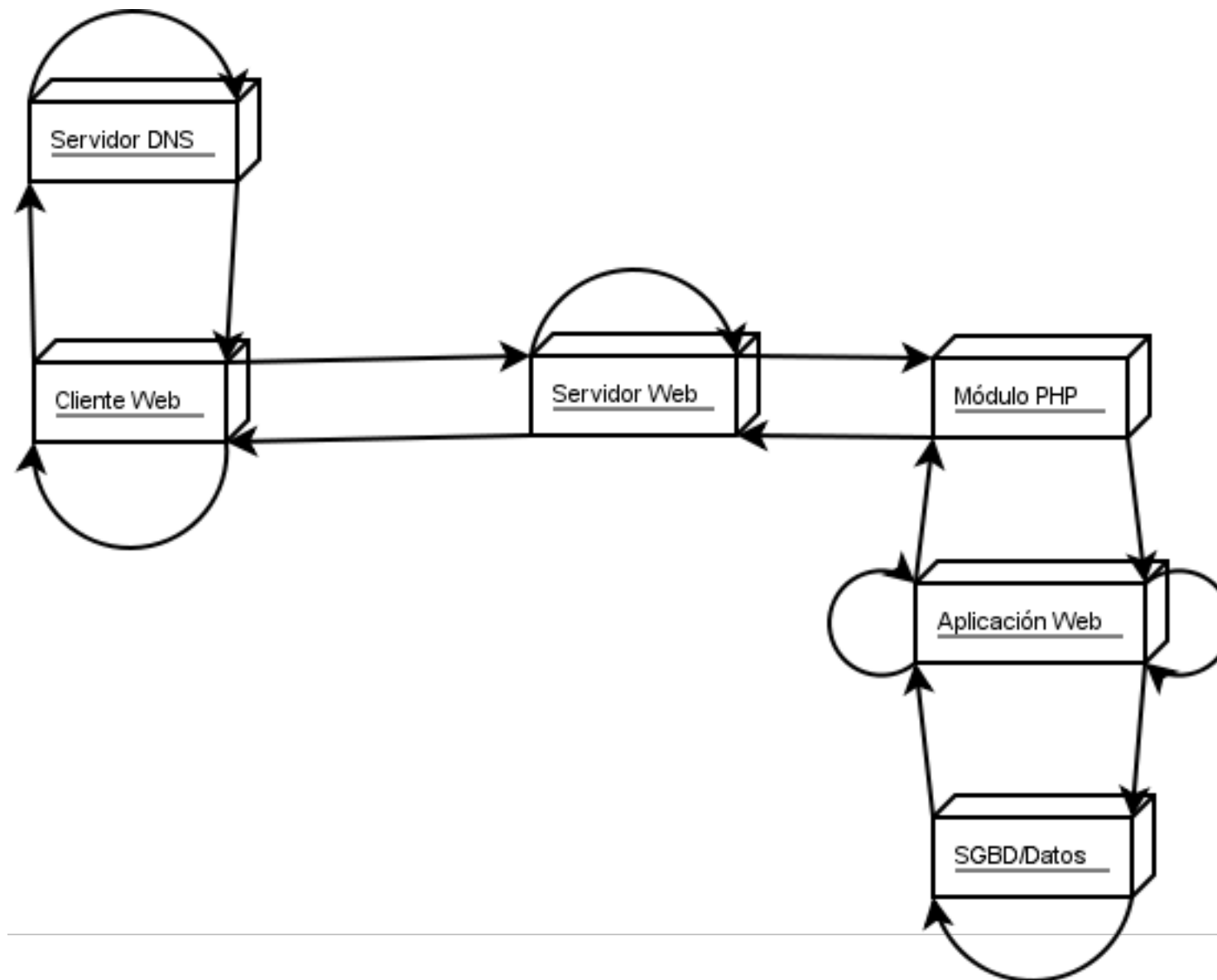


Resumen de una respuesta web

- Al igual que la petición, una respuesta HTTP contiene piezas de información adicionales conocidas como cabeceras HTTP o headers. Por ejemplo, una importante cabecera de respuesta HTTP es Content-Type. El cuerpo o body de un mismo recurso puede ser devuelto en múltiples formatos diferentes como HTML, XML o JSON y la cabecera Content-Type utiliza Internet Media Types como texto/html para decirle al cliente qué formato está devolviendo.



Proceso de una Petición Web



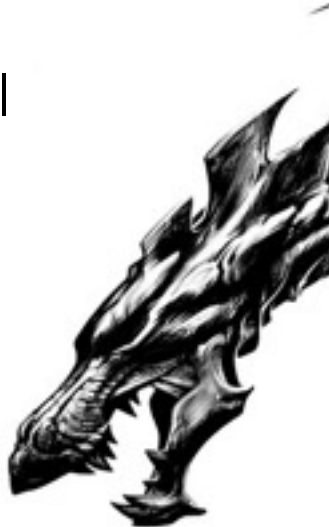
Proceso de una petición web

- Cliente Web: Solicitud de conversión del nombre de una máquina al Servidor DNS. Por ejemplo: `www.google.es`
- Servidor DNS: Recepción y tratamiento de la solicitud.
- Servidor DNS: Una vez recibida la petición realiza las consultas necesarias para resolver y obtener la dirección IP.
- Servidor DNS: Devuelve la dirección IP que corresponde al Servidor Web al navegador Web.
- Cliente Web: Realiza la solicitud de información mediante una URL(Método GET) o un formulario (Método POST). Dicha solicitud incluye: la dirección IP del servidor web, el puerto del servidor web, URL y parámetros POST(esta sólo en el caso de formularios normalmente).
- Servidor Web: Recibe la petición realiza el Control de Acceso, Análisis de la petición y localización del recurso.
- Servidor Web: Selección del Servidor de Aplicaciones para interpretar la ruta elegida y redirección de la petición.



Proceso de una petición web

- Servidor de aplicaciones: Recibe la petición y mediante la URL elige la ejecución del programa elegido (Módulo de PHP por ejemplo) y pasa el control a la aplicación web.
- Aplicación Web: La aplicación web recibe la petición y procesado de los parámetros GET o POST y elige la funcionalidad elegida.
- Aplicación Web: Realización de las llamadas SQL al Servidor de SGBD.
- SGBD: Recogida de la SQL
- SGBD: Procesado de la petición SQL y obtiene los resultados
- SGBD: Retorno de los datos solicitados a la aplicación web.
- Aplicación Web: Procesado de los datos devueltos por el SGBD y Generación del HTML resultante.
- Aplicación Web: Devuelve los resultados al Servidor de aplicaciones



Proceso de una petición web

- Servidor de aplicaciones: Procesa los datos de la aplicación Web
- Servidor de Aplicaciones: Pasa el HTML resultante a Apache.
- Apache: Devuelve el HTML al cliente Web.
- Cliente Web: Presentación del HTML resultante
- Cliente Web: Descarga los CSS y elementos relacionados
- Cliente Web: Ejecución del Javascript Inicial
- Cliente Web: Entra en el bucle principal de eventos
- Cliente Web: Ejecuta el Javascript cuando termina de cargar la página



Conclusiones

Aquí tenemos la
introducción inicial al
protocolo http, incluido el
acceso a peticiones web,
simples y complejas

<http://cursosdedesarrollo.com/>



Datos de Contacto

<http://www.cursosdedesarrollo.com>
info@cursosdedesarrollo.com

<http://cursosdedesarrollo.com/>



Licencia



David Vaquero Santiago

Esta obra está bajo una
Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0
Internacional

Derivada de:

<http://www.arkaitzgarro.com/xhtml/>

y

<http://javiereguiluz.com/>

<http://cursosdedesarrollo.com/>

