

Tema 1: Arquitectura Web.

1. Introducción.
 1. Componentes semánticos de la Web.
 - URI
 - HTTP
 - HTML
 2. Componentes software de la WEB (Arquitectura Web).
 1. Cliente
 2. Servidor
 3. Proxy
 3. Arquitectura de las Aplicaciones Web

ARS - 2011

Arquitectura Web

1

1. Introducción.

La Web.

- World Wide Web:
 - Universo de información interconectada, accesible a través de internet.
- Propuesta por Tim Berners-Lee (1989).
- Ha tenido mayor difusión que otros servicios contemporáneos (Archie, Gopher, WAIS), gracias sobre todo a uno de sus elementos: el HTML.
 - Hipertexto: Lectura no secuencial de docs, basada en asociación de ideas.
 - Hipermedia: Hipertexto + gráficos, sonidos, vídeos etc.
- En 2001 había más de un billón de URLs accesibles al público, repartidas entre más de 30 millones de servidores. En 2007 155 millones de páginas en Internet.

ARS - 2011

Arquitectura Web

2

1. Introducción.

1.1 Componentes semánticos de la Web

- **URI: Uniform Resource Identifier.**
 - Identifica los recursos web para su acceso y manipulación.
- **HTML: HyperText Markup Language.**
 - Lenguaje de marcas.
 - Provee una representación estándar de los documentos hipertexto en formato ASCII.
 - Permite formatear texto, integrar imágenes, referenciar otros documentos, etc.
- **HTTP: Hypertext Transfer Protocol.**
 - Protocolo que permite a los componentes web (cliente, servidores, etc) comunicarse de una forma estándar y bien definida.
 - Define el formato y el significado de los mensajes intercambiados entre componentes web.

1. Introducción.

1.1 Componentes semánticos de la Web.

Codificación URI

- Identifica de forma única el recurso.
- 2 Tipos:
 - **URN: Uniform Resource Name.**
 - Identifica de forma única el recurso, independientemente de donde resida (RFC 2141).
 - El mismo recurso situado en máquinas diferentes poseen el mismo identificador.
 - Todavía está en fase experimental.
 - **URL: Uniform Resource Locator.**
 - La forma más común de identificar el recurso.
 - Señala exactamente donde se encuentra el recurso.
 - 3 partes principales:
 - Protocolo o esquema + servidor + nombre del recurso.

1. Introducción.
1.1 Componentes semánticos de la Web.

URL

■ **Sintaxis:**

- **esquema:** `//[usuario];[password]@<maquina>:[puerto]/<camino>[parametros]?[consulta]#[sección]`
- **Esquema:** protocolo (http, https, file, ftp, news, mailto, ..).
- **Usuario:password:** para recursos de acceso restringido
- **Máquina:** nombre del servidor
- **Puerto:** número del puerto donde escucha el servidor.
- **Camino:** Directorio virtual y nombre del recurso.
- **Parámetros:** pares nombre=valor utilizados por algunos esquemas.
- **Consulta:** pares nombre=valor separados por &, utilizados en algunas aplicaciones web.
- **Sección:** nombre de una parte del recurso.

■ **Ejemplos:**

- `http://www.hardware.com:2000/pc/check.cgi?item=1273&model=B`
- `ftp://jose:suclave@www.hardware.com/informacion.txt`

ARS - 2011

Arquitectura Web

5

1. Introducción.
1.1 Componentes semánticos de la Web.

Conjunto de caracteres de la URL

- La URL ha sido diseñada para ser portable.
- Los caracteres especiales incluidos en la URL son transformados antes de ser enviados:
 - los caracteres de letras y números de la tabla ASCII estándar se dejan intactos.
 - Los espacios en blanco son sustituidos por +.
 - Los caracteres especiales son sustituidos por su valor hexadecimal con el prefijo '%'
 - Los caracteres con un significado especial ("+", "&", "=", ",", "/", "?", "#", etc.) son también sustituidos por su código hexadecimal.

Núñez & Cía. → `N%FA%F1ez+%26+C%EDa.`

ARS - 2011

Arquitectura Web

6

2. Componentes software de la Web.

Arquitectura Cliente/Servidor

- El modelo cliente-servidor es una arquitectura software que involucra uno o más clientes solicitando servicios a uno o más servidores.
- El cliente puede ser un proceso corriendo en un computadora o en un dispositivo como una PDA o un teléfono móvil.
- El servidor puede ser un proceso corriendo en un computadora (normalmente de altas prestaciones) o en varias (sistemas distribuidos).
- En la arquitectura Web actual aparecen además elementos que se sitúan en medio (proxies, cachés)
- Beneficios:
 - Usabilidad/flexibilidad/interoperabilidad/ escalabilidad.

ARS - 2011

Arquitectura Web

7

2. Componentes software de la Web.

Arquitectura Cliente/Servidor



Arquitectura cliente-servidor tradicional



Arquitectura cliente-servidor interactiva para la WEB

ARS - 2011

Arquitectura Web

8

2. Componentes software de la Web.

2.1 Clientes

- Originan el tráfico web.
 - Envían las peticiones y reciben las respuestas, interpretándolas y mostrándolas.
- Dos clases de clientes web: navegadores y robots.
- Los navegadores (Netscape, IE, etc).
 - Las peticiones están dirigidas por el usuario.
 - Repiten peticiones al mismo objeto cuando navegan por un site.
 - Utilizan caches de memoria y disco.
- Robots (spiders, y agentes inteligentes).
 - Las peticiones son automatizadas.
 - La velocidad y carga está limitada por la velocidad de proceso, y por la velocidad de la red.

ARS - 2011

Arquitectura Web

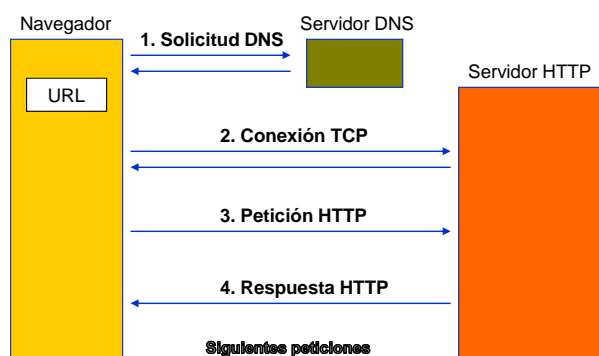
9

2. Componentes software de la Web.

2.1 Clientes

Navegadores

- Programa que realiza las peticiones, a solicitud de un usuario, y recibe, analiza y presenta las respuestas.
- Pasos:



ARS - 2011

Arquitectura Web

10

2. Componentes software de la Web.

2.1 Clientes

Funciones de los navegadores

- Construyen y mandan la petición HTTP
- Reciben, interpretan y presentan la respuesta.
 - La apariencia final depende de los parámetros de configuración
 - Algunos recursos precisan aplicaciones de ayuda para visualizarse
 - Código MIME
- Proporcionan el interfaz para conectarse y utilizar otros servicios: mail, news, ftp, etc.
 - El protocolo por defecto es http.
- Caché local.
 - Sirve recursos guardados en la caché sin conectarse al servidor.
 - Consistencia (asegurar que el recurso guardado es aún válido antes de visualizarlo):
 - Fuerte: revalida siempre el recurso conectándose al servidor.
 - Débil: se basa en los parámetros HTTP, y en los parámetros de configuración, para decidir si es necesario revalidar el recurso.
- Manejo de las Cookies.

IST -2010

Arquitectura Web

11

2. Componentes software de la Web.

2.1 Clientes

Spiders

- Robots dedicados a la búsqueda automática de información.
 - Parten de la página principal de un sitio web, y examinan los enlaces embebidos que encuentran.
 - Las peticiones están espaciadas en el tiempo para no sobrecargar el servidor.
- La información se utiliza posteriormente en aplicaciones de búsqueda (google, yahoo).
- Los recursos dinámicos (CGI, PHP, etc.) no son indexados (normalmente tienen parametros que el spider ignora).
- Algunos sitios web no desean ser indexados:
 - Controlan el acceso de robots.
 - Los recursos HTML incluyen una directiva META:
 - `<META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">`

ARS - 2011

Arquitectura Web

12

2. Componentes software de la Web.

2.2 Servidores Web

- Programa que contesta y genera la respuesta HTTP a las peticiones de recursos web por parte del cliente
 - Involucra múltiples servidores, scripts, bases de datos, ..
- Trabajo básico:
 - Se conecta con el cliente.
 - Recibe el mensaje HTTP de la petición.
 - Procesa el mensaje HTTP.
 - Localiza y envía el resultado (en forma de mensaje HTTP)
- Los servidores de altas prestaciones, además:
 - Tratan múltiples peticiones:
 - hilos para manejar cada conexión.
 - Generan dinámicamente contenido: ASP, PHP, JSP
 - Caché.
- Los más populares son Apache e IIS.

ARS - 2011

Arquitectura Web

13

2. Componentes software de la Web.

2.2 Servidores Web

Manejo de las peticiones

- Los servidores proveen acceso a los recursos:
 - Estáticos.
 - Dinámicos (scripts que generan la respuesta dinámicamente).
- Pasos:
 - Lee e interpreta el mensaje de petición (método, URL, cookies,..)
 - Localiza físicamente el recurso apuntado en la URL
 - Determina si el cliente está autorizado (controla el acceso)
 - Genera el mensaje de respuesta y lo transmite (cookies)
 - Registra la operación en un fichero de log
- Controlan el acceso a recursos restringidos:
 - Autenticación.
 - Piden al usuario que se identifique (login y password)
 - La información se incluye en la cabecera del mensaje.
 - Autorización.
 - Comprueba en su lista de acceso si el usuario está autorizado.

ARS - 2011

Arquitectura Web

14

2. Componentes software de la Web.

2.2 Servidores Web

Recursos dinámicos

- El contenido se crea tras recibir la petición.
 - El recurso apuntado en la URL incluye código que debe ser ejecutado para resolver el contenido de la respuesta.
- Dos tipos:
 - Scripts de servidor (PHP, ASP, JSP)
 - Ficheros HTML (con extensiones diversas) que incluyen macros que el servidor (un módulo) interpreta para insertar la información precisa.
 - El servidor reconoce este tipo de recursos por la extensión
 - Programas independientes (CGI, Servlets)
 - Programa separado del servidor que genera la respuesta (el servidor invoca el programa tras la petición del cliente)
 - Fórmulas:
 - Procesos separados (CGI). Pueden ser persistentes (FastCGI).
 - Módulos software en el mismo proceso (Servlets).

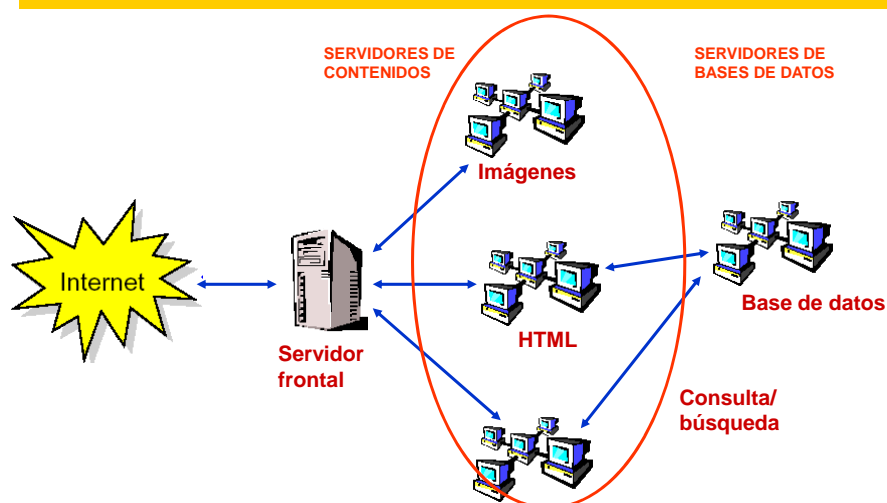
ARS - 2011

15

2. Componentes software de la Web.

2.2 Servidores Web

Servidores complejos (I)



ARS - 2011

Arquitectura Web

16

2. Componentes software de la Web.

2.2 Servidores Web

Servidores complejos (II)

- Algunos sitios web tienen un conjunto amplio de máquinas servidoras actuando como un único servidor.
 - Cluster de máquinas o granja de servidores.
 - Estructuradas en capas.
 - No son necesariamente servidores virtuales (múltiples sitios web soportados por la misma máquina).
- Servidor frontal:
 - Caché, direcciona las peticiones, balancea la carga
- Servidores de contenido:
 - Algunas veces diferenciados de acuerdo al tipo de petición (HTML, imágenes, consulta, búsqueda).
- Servidores de bases de datos.
 - Información que contribuye a la parte dinámica.

ARS - 2011

Arquitectura Web

17

2. Componentes software de la Web.

2.3 Proxies

Intermediarios Web

- **Proxy**
 - Programa que actúa a la vez como servidor y cliente, situándose en medio de los dos.
 - Las peticiones son servidas internamente o pasadas a otro servidor. Actúan filtrando peticiones a servidores.
- **Pasarela**
 - Un servidor que actúa como intermediario de algún otro servidor.
 - A diferencia del proxy, una pasarela proxy recibe peticiones como si fuera el servidor original.
 - El cliente no tiene que preocuparse por el hecho de que esté comunicándose con la pasarela.
 - Actúan de intermediarios a otros sistemas no-http.
- **Túnel**
 - Programa intermediario que actúa como un túnel no controlado, permitiendo tráfico no-HTTP a través de conexiones HTTP
 - Un vez activo, un túnel no es considerado como parte de la comunicación HTTP, aunque haya sido activado por la petición HTTP.
 - El túnel deja de existir cuando la comunicación se cierra y no cachea las respuestas..

ARS - 2011

Arquitectura Web

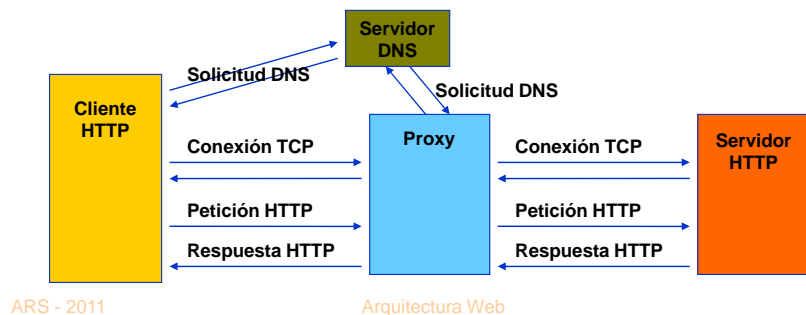
18

2. Componentes software de la Web.

2.3 Proxies

El proxy

- Intermediario web situado entre el cliente y el servidor para reducir comunicación no deseada.
 - A veces forma parte de los mecanismo de seguridad.
 - La mayoría de veces actúan como caches.
- Realizan simultáneamente el papel de servidor y cliente:



19

2. Componentes software de la Web.

2.3 Proxies

Funciones del proxy

- **Caché:**
 - Guarda las respuestas.
 - Comparte accesos con otros clientes.
- **Filtra las peticiones y las respuestas (seguridad).**
 - Filtra las peticiones a ciertos sitios web.
 - Filtra respuestas (pe. Basándose en el tamaño)
- **Guardan el anonimato.**
 - Los servidores no conocen la IP del cliente. Privacidad del cliente.
- **Transforman las peticiones y las respuestas.**
 - Puede transformar la petición para ajustarse a capacidades que comparte con el servidor (pe. Algoritmos de compresión).
 - Puede ajustar la respuesta a las capacidades del cliente.
- **Actúan también de pasarelas a otros sistemas no-HTTP.**

ARS - 2011

Arquitectura Web

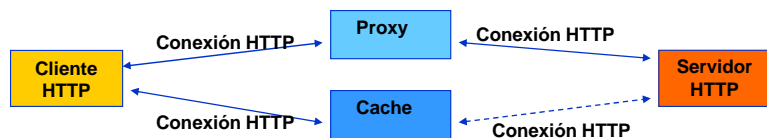
20

2. Componentes software de la Web.

2.3 Proxies

Cachés

- Cuando actúan como caches gestionan las peticiones en la parte de los clientes.
 - El navegador envía la petición a la caché.
 - La Caché devuelve el recurso si está almacenado localmente y está actualizado.
 - Basado en los campos URL, ETag, fecha.
 - En caso contrario, la caché pasa la solicitud al servidor
 - Si está pasado de fecha, la caché valida el recurso
- Diferencia con el resto de proxies:
 - Algunas peticiones no llegan al servidor origen.



ARS - 2011

Arquitectura Web

21

2. Componentes software de la Web.

2.3 Proxies

Beneficios de las cachés

- Los Proxies que actúan como caches explotan las peticiones anteriores de un grupo de clientes.
- Las cachés benefician a clientes, a servidores, y a la red.
 - **Ancho de banda:** Reducen la utilización de la red
 - Era el objetivo inicial de las caches.
 - Especialmente útiles en entornos con restricciones en el ancho de banda (pe. en enlaces internacionales)
 - **Velocidad:** Reducen el tiempo de espera.
 - Cuanto más próximo está la caché del cliente, más rápido es el tiempo de respuesta.
 - **Carga del servidor:** Partes de las peticiones son resueltas por la caché.
- Empíricamente, las caches mejoran en un 50% la velocidad de respuesta y reducen en un 40% la cantidad de bytes transmitidos.

ARS - 2011

Arquitectura Web

22

Aplicaciones Web

- Una aplicación web es proporcionada por un servidor web y utilizada por usuarios que se conectan desde cualquier punto vía clientes web (navegadores).
- Definición:
 - Son aplicaciones basadas en el modelo Cliente/Servidor que gestionan servidores web, y que utilizan como interfaz páginas web.
 - La colección de páginas son en una buena parte dinámicas (ASP, PHP, etc.), y están agrupadas lógicamente para dar un servicio al usuario.
 - El acceso a las páginas está agrupado también en el tiempo (sesión: periodo de tiempo en que un cliente hace uso continuado de una aplicación).
 - Ejemplos: venta de libros, reserva de billetes, etc.

Componentes de una aplicación Web

- Lógica de negocio.
 - Parte más importante de la aplicación.
 - Define los procesos que involucran a la aplicación.
 - Conjunto de operaciones requeridas para proveer el servicio.
- Administración de los datos.
 - Manipulación de BD y archivos.
- Interfaz:
 - Los usuarios acceden a través de navegadores, móviles, PDAs, etc.
 - Funcionalidad accesible a través del navegador.
 - Limitada y dirigida por la aplicación.

3. Arquitectura de las aplicaciones Web

Modelo de capas

- Las aplicaciones web se modelizan mediante lo que se conoce como modelo de capas.
 - Una capa representa un elemento que procesa o trata información.
- Tipos:
 - Modelo de dos capas: La información atraviesa dos capas entre la interfaz y la administración de los datos.
 - Modelo de n-capas: La información atraviesa varias capas.
 - El más habitual es el modelo de tres capas.

ARS - 2011

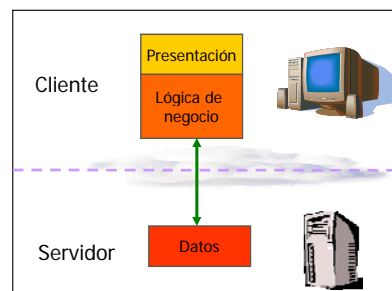
Arquitectura Web

25

3. Arquitectura de las aplicaciones Web.

Modelo de dos Capas.

- Gran parte de la aplicación corre en el lado del cliente (fat client).
- Capas:
 - **Cliente (fat client):**
 - La lógica de negocio está inmersa dentro de la aplicación que realiza el interfaz de usuario, en el lado del cliente.
 - **Servidor:**
 - Administra los datos.
- Limitaciones.
 - Es difícilmente escalable
 - Número de conexiones reducida
 - Alta carga de la red.
 - La flexibilidad es restringida
 - La funcionalidad es limitada.



ARS - 2011

Arquitectura Web

26

3. Arquitectura de las aplicaciones Web.

Modelo de tres Capas (I)

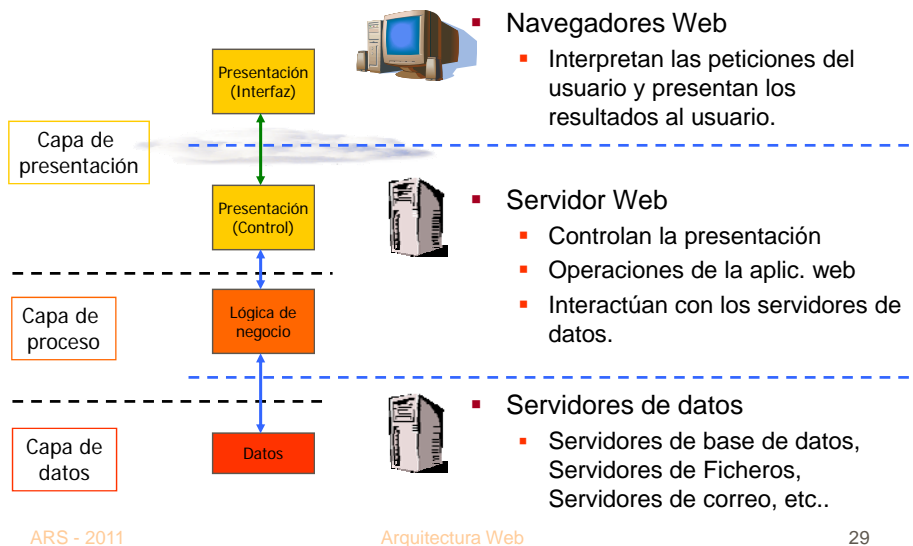
- Diseñada para superar las limitaciones de las arquitecturas ajustadas al modelo de dos capas
- Introduce una capa intermedia (la capa de proceso) entre presentación y los datos
 - Los procesos pueden ser manejados de forma separada al interfaz de usuario y a los datos
 - La capa intermedia centraliza la lógica de negocio, haciendo la administración más sencilla.
- Pueden integrar datos de múltiples fuentes
- Las aplicaciones web actuales se ajustan a este modelo.

3. Arquitectura de las aplicaciones Web.

Modelo de tres Capas (II)

- Capas:
 - Capa de presentación (parte en el cliente y parte en el servidor)
 - Recoge la información del usuario y la envía al servidor (cliente)
 - Manda información a la capa de proceso para su procesamiento
 - Recibe los resultados de la capa de proceso
 - Generan la presentación
 - Visualizan la presentación al usuario (cliente)
 - Capa de proceso (servidor web)
 - Recibe la entrada de datos de la capa de presentación
 - Interactúa con la capa de datos para realizar operaciones
 - Manda los resultados procesados a la capa de presentación
 - Capa de datos (servidor de datos)
 - Almacena los datos
 - Recupera datos
 - Mantiene los datos
 - Asegura la integridad de los datos

Modelo de tres Capas (III)



APIs y tecnologías que forman parte de Java EE

