

Grupo 2

HTTP-PROXY-CACHING

- *Scollo Denise.*
- *Lescano Oscar.*
- *Martín Manuel.*

HTTP

HTTP

Hypertext Transfer Protocol

- ◉ Protocolo utilizado en cada transacción de la World Wide Web.
- ◉ Desarrollado por World Wide Web Consortium y IETF (Internet Engineering Task Force).
- ◉ La colaboración culminó en 1999 con la publicación del RFC 2616, que especifica la versión 1.1

HTTP

- ◉ Define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web para comunicarse.
- ◉ Protocolo orientado a transacciones, sigue el esquema petición respuesta entre cliente - servidor.
- ◉ Al cliente se le conoce como “User Agent” (Navegador Web). A la información transmitida se la llama “Recurso”.
- ◉ Los recursos se identifican mediante un URL

HTTP

- Es un protocolo sin estado (No guarda información sobre conexiones anteriores)
- El desarrollo web necesita mantener estado. Para esto se utilizan las cookies.



HTTP

Cookies

- ⦿ Pequeña información enviada por un sitio web y almacenada en el navegador del usuario.
 - > Lleva el control de los usuarios.
 - > Consigue información sobre los hábitos de navegación del usuario.

HTTP

Transacciones

- ⦿ Esta formada por:

- > *Encabezado:*

- Bloque de datos que precede a la información.
 - Le da flexibilidad al protocolo.
 - Especifica la acción requerida del servidor, el tipo de dato retornado o el código de estado.

- > Línea en blanco.

- > Datos.

HTTP

Transacciones

⦿ HTTP_ACCEPT:

- > Especifica los tipos de archivos que el cliente aceptará.
- > Los elementos deben estar separados por una coma.
 - Ejemplo: HTTP: tipo, tipo.

⦿ HTTP_USER_AGENT:

- > Navegador que se utiliza para realizar la petición.
 - Formato: Software/Versión – Biblioteca/Versión

HTTP

Transacciones

- ◉ El servidor envía al cliente:
 - > Código de estado.
 - > Información. HTTP permite enviar archivos de todo tipo.
 - > Información sobre el objeto que se retorna.

HTTP

Códigos de estado

- ◉ Dan una descripción corta del estado de la petición.
 - > 1xx Mensajes.
 - > 2xx Conexión exitosa.
 - > 3xx Redirecciones.
 - > 4xx Errores del cliente.
 - > 5xx Errores del servidor.

HTTP

Códigos de estado

- ◉ 1xx:
 - > 100 – 111 Conexión rechazada.
- ◉ 2xx:
 - > 200 OK.
 - > 204 Sin Contenido.
- ◉ 3xx:
 - > 302 Encontrado
 - > 305: Utilice Proxy

HTTP

Códigos de Estado

● 4xx:

- > 401 No Autorizado.
- > 403 Prohibido.
- > 404 No Encontrado.

● 5xx:

- > 500 Error Interno
- > 503 Servicio No Disponible

Error 404

Page Not Found

Mon Jul 14 2008 4:43:30 am EDT

168.192.104.2 tried to load www.revacsystems.com/images/

User Agent = Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; InfoPath.1)

You are in a Wrong page!

Visit

[Home Page](#)



HTTP

Métodos de Petición

- ◉ Define 8 métodos – Indica la acción a efectuar sobre el recurso.
 - > Get: Pide una representación del recurso especificado
 - > Head: Idéntico al “Get” sin el cuerpo de la repuesta.
 - > Post: Envía los datos para que sean procesados por el recurso identificado.

HTTP

Métodos de Petición

- > Put: realiza un upload de un recurso especificado (archivo).
- > Delete: Borra el recurso.
- > Trace: Se utiliza con fines de comprobación y diagnóstico.
- > Options: Devuelve los métodos HTTP que el servidor soporta para un URL.
- > Connect: Se utiliza para saber si se tiene acceso a un host.

HTTP

Ejemplos de diálogo.

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: nombre-cliente
[Línea en blanco]
```

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221

<html>
<body>
<h1>Página principal de tuHost</h1>
  (Contenido)
  .
  .
  .
</body>
</html>
```

HTTP

Versiones

◉ 0.9

- > Obsoleta
- > No especifica su Versión.
- > Soporta solo el comando GET.
- > El cliente no puede enviar mucha información.

◉ HTTP/1.0 (1996)

- > Primera revisión del protocolo.
- > Especifica su Versión.
- > Se utiliza sobre todo en servidores proxy.

HTTP

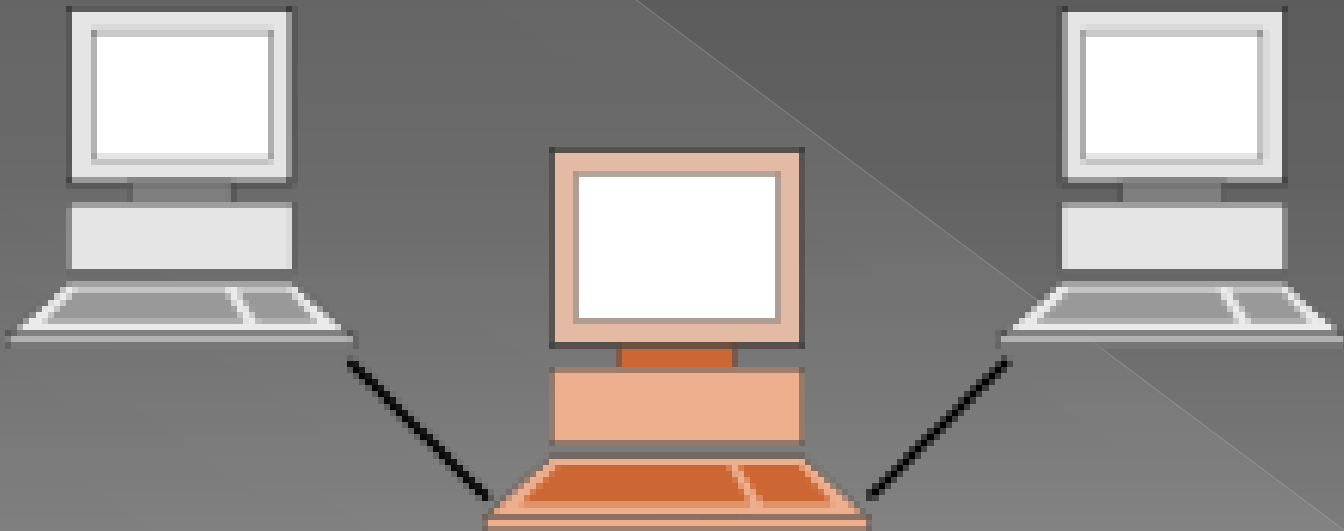
Versiones

- HTTP/1.1 (1999)
 - > Versión Actual.
 - > Conexiones persistentes activadas por defecto.
 - > Enviar múltiples peticiones a la vez.
 - > Funciona bien con los proxies.

PROXY

PROXY

- Programa o dispositivo que sirve de intermediario en peticiones de recursos.



PROXY

Características generales

- ⦿ Proporciona cache.
- ⦿ Registro de trafico.
- ⦿ Control de accesos.
- ⦿ Mejorar rendimiento.
- ⦿ Prohibir cierto tipo de tráfico.
- ⦿ Mantiene el anonimato.

PROXY

- Proxy Local:

- > Quien implementa la política es el mismo que hace la petición.

- Proxy Externo:

- > Quien implementa la política es una entidad externa.

PROXY

◎ Proxy Caché:

- > Conserva el contenido solicitado.
- > Menos consumo en el ancho de banda.
- > Acelera la respuesta en futuras peticiones.
- > Conecta con servidor remoto para chequear que la versión de caché sea la misma.

PROXY

- ◉ Proxy Transparente:

- > Combina servidor proxy con un Firewall.
- > Las conexiones son desviadas hacia el proxy.
- > Normalmente el usuario no conoce su existencia.
- > Utilizado por empresas proveedoras de Internet.

PROXY

- ◎ Proxy Inverso (Reverse Proxy):
 - > Situado en el alojamiento de uno o mas servidores web.
 - > Tráfico procedente y con destino a algunos de los servidores web es recibido por el Proxy.

PROXY

Proxy Inverso

> Razones:

- Seguridad: Es una capa adicional de defensa.
- Cifrado/Aceleración SSL: Cuando se crea un sitio web seguro, el cifrado SSL no lo hace el mismo servidor web.
- Distribución de Carga: Puede distribuir la carga entre varios servidores web.
- Caché de Contenido Estático: Puede almacenar contenido estático.

PROXY

- ◎ Proxy NAT / Enmascaramiento:
 - > Reescribe direcciones fuente/destinos de paquetes IP.
 - > Encargado de traducir direcciones privadas a la única dirección pública.
 - > Su función reside en los Firewalls.
 - > No hay conexión directa entre el exterior y la red privada.

PROXY

● Proxy Abierto:

- > Acepta peticiones desde cualquier ordenador.
- > Se utiliza como pasarela para envío masivo de correo de spam.
- > Muchos servidores/corremos le niegan el acceso a sus servidores (“Blacklist”).

PROXY

- Cross-Domain Proxy:

- > Usado por tecnologías web asíncronas.
- > Permite establecer conexiones entre elementos ubicados en distintos dominios.
- > Recibe la petición y la reenvía al dominio externo.

CONCLUSIONES

HTTP

- ◉ Versión Actual: HTTP/1.1
- ◉ Define sintaxis y semántica en la arquitectura web.
- ◉ Protocolo sin Estado. Por esto se utilizan las cookies.
- ◉ Transacción formada por un encabezado y datos.
- ◉ Define 8 métodos de petición.
- ◉ Ideal para transferencia de archivos multimedia.

PROXY

Ventajas

- Control: Se pueden limitar/restringir los derechos de los usuarios.
- Ahorro: Sólo un usuario (proxy) necesita los recursos necesarios para esa funcionalidad.
- Velocidad: Puede hacer caché.
- Filtrado: Puede negarse a responder peticiones.
- Modificación: Puede falsificar información o modificarla.

PROXY

Desventajas

- Anonimato: Si todos se identifican como uno solo, es difícil que el recurso los diferencie.
- Abuso: Al estar dispuesto a recibir muchas peticiones, puede que haga algún trabajo que no toque.
- Carga: Un proxy hace el trabajo de muchos usuarios.
- Intromisión: Algunos usuarios no quieren pasar por el proxy y menos si hace de caché, que guarda datos.

GRACIAS