

# Planificación y Administración de Redes: HyperText Transfer Protocol



IES Gonzalo Nazareno  
**CONSEJERÍA DE EDUCACIÓN**

Jesús Moreno León  
Raúl Ruiz Padilla  
j.moreno1@gmail.com  
Septiembre 2010

---

Estas diapositivas son una obra derivada de las transparencias  
del Grupo de Sistemas y Comunicaciones  
de la Universidad Rey Juan Carlos  
Puede encontrarse una versión de este documento en  
<http://gsyc.es/moodle>

© Jesús Moreno León, Raúl Ruiz Padilla, Septiembre de 2010

Algunos derechos reservados.  
Este artículo se distribuye bajo la licencia  
“Reconocimiento-CompartirIgual 3.0 España” de Creative  
Commons, disponible en  
<http://creativecommons.org/licenses/by-sa/3.0/es/deed.es>

Este documento (o uno muy similar)  
está disponible en (o enlazado desde)  
<http://informatica.gonzalonazareno.org>

# ¿Qué es HTTP?

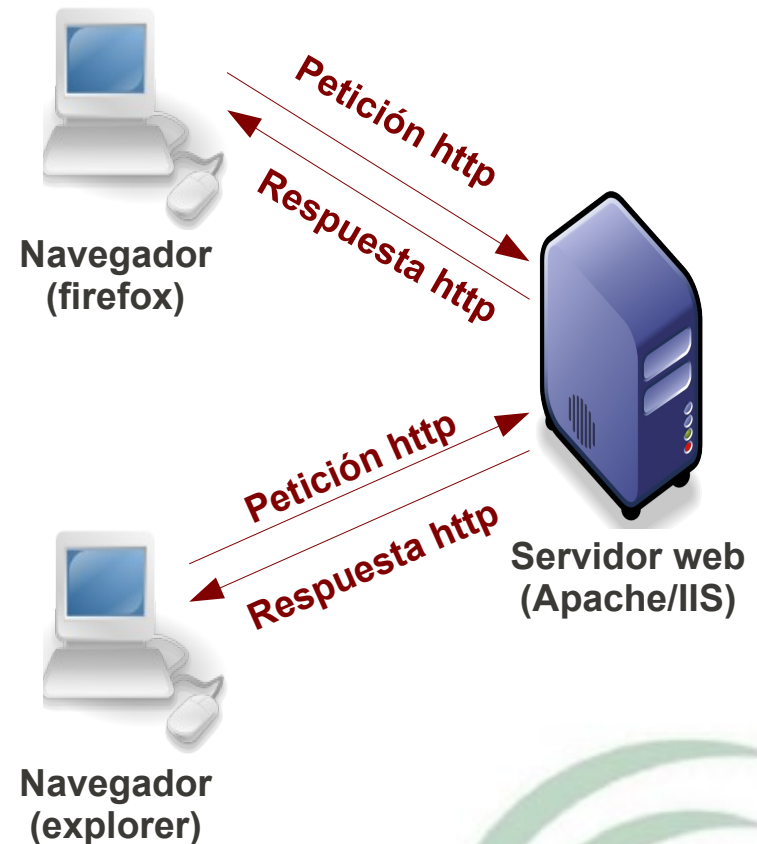
---

- Protocolo utilizado para servir páginas web
- Protocolo cliente-servidor:
  - Cliente: navegador web que pide páginas y, al recibirlas, las muestra al usuario
  - Servidor: servidor web en el que están alojadas páginas que piden los clientes
- Habitualmente funciona sobre TCP, con el servidor (normalmente) en el puerto 80
- HTTP puede servir tanto contenido estático (ficheros) como dinámico (el resultado de ejecutar programas en el servidor)



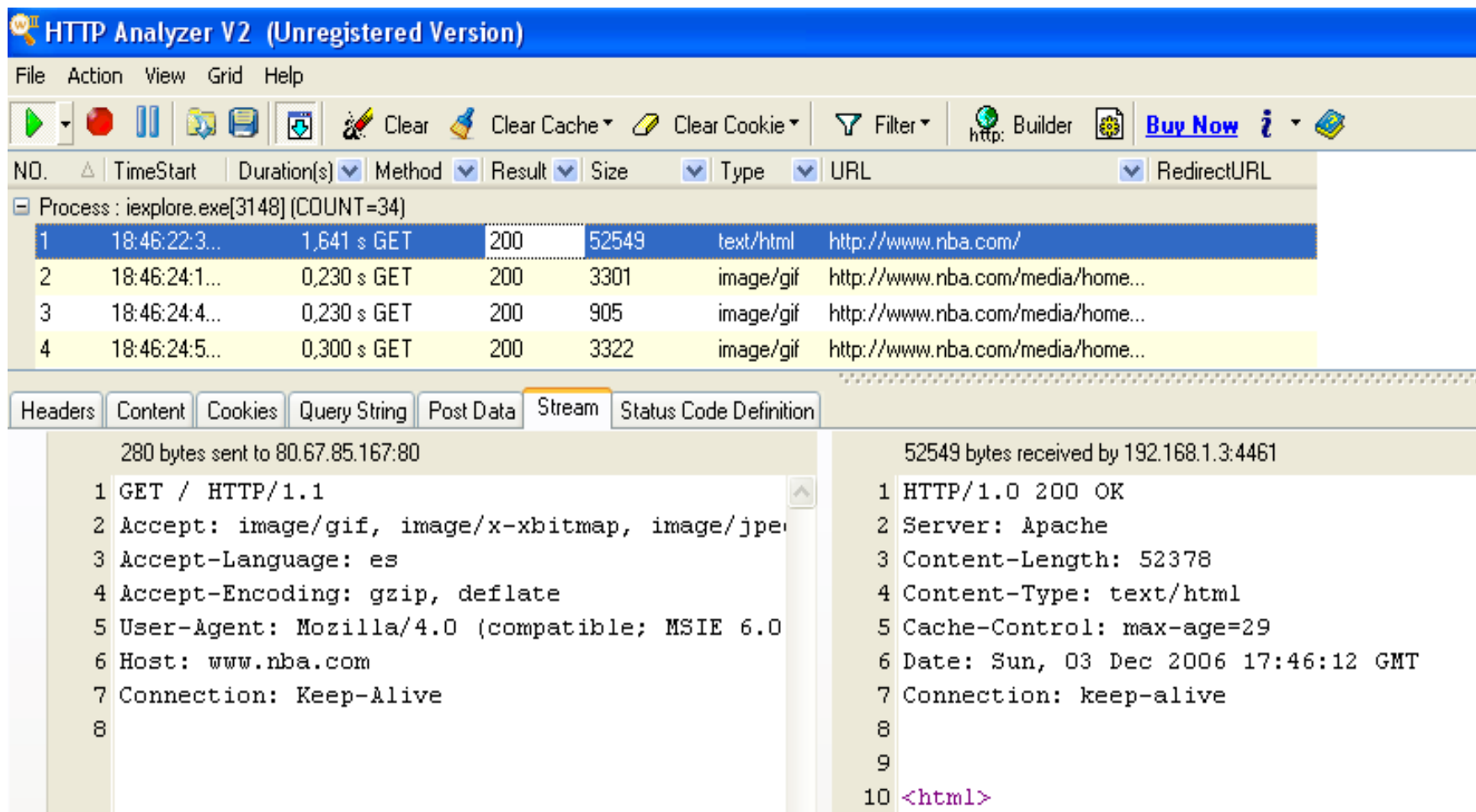
# Interacción cliente-servidor

- El cliente abre una conexión TCP con el servidor
- El cliente envía un mensaje de petición
- El servidor responde con un mensaje de respuesta
- El servidor cierra la conexión TCP
- **HTTP no mantiene estado** (no hay información sobre las conexiones entre una petición y otra)



# Formato de los mensajes

- Mensajes legibles (son líneas de texto *casi en inglés*) compuesto por línea inicial, cabeceras y cuerpo:



The screenshot displays the HTTP Analyzer V2 (Unregistered Version) interface. The top menu bar includes File, Action, View, Grid, and Help. Below the menu is a toolbar with various icons for actions like Run, Stop, Clear, and Filter. The main window shows a list of HTTP requests captured from the process iexplore.exe[3148]. The list has columns for NO., TimeStart, Duration(s), Method, Result, Size, Type, URL, and RedirectURL. The first request is highlighted in blue.

NO.	TimeStart	Duration(s)	Method	Result	Size	Type	URL	RedirectURL
1	18:46:22:3...	1,641 s	GET	200	52549	text/html	http://www.nba.com/	
2	18:46:24:1...	0,230 s	GET	200	3301	image/gif	http://www.nba.com/media/home...	
3	18:46:24:4...	0,230 s	GET	200	905	image/gif	http://www.nba.com/media/home...	
4	18:46:24:5...	0,300 s	GET	200	3322	image/gif	http://www.nba.com/media/home...	

Below the list, the 'Stream' tab is selected, showing the raw HTTP data. The left pane displays the request sent to 80.67.85.167:80, and the right pane displays the response received by 192.168.1.3:4461.

**Request (280 bytes sent to 80.67.85.167:80):**

```
1 GET / HTTP/1.1
2 Accept: image/gif, image/x-xbitmap, image/jpeg
3 Accept-Language: es
4 Accept-Encoding: gzip, deflate
5 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0
6 Host: www.nba.com
7 Connection: Keep-Alive
8
```

**Response (52549 bytes received by 192.168.1.3:4461):**

```
1 HTTP/1.0 200 OK
2 Server: Apache
3 Content-Length: 52378
4 Content-Type: text/html
5 Cache-Control: max-age=29
6 Date: Sun, 03 Dec 2006 17:46:12 GMT
7 Connection: keep-alive
8
9
10 <html>
```

# Formato de los mensajes

---

- Peticiones:
  - La línea inicial especifica el recurso que se solicita y qué se quiere de él
    - Nombre de método (`GET`, `POST`, `HEAD`)
    - Camino de acceso (`path`)
    - Versión de HTTP (`HTTP/x.x`)

Ejemplo:

`GET /directorio1/directorio2/fichero.html HTTP/1.1`



# Formato de los mensajes

---

- Respuestas:
  - La línea inicial proporciona información del estado
    - Versión de HTTP (HTTP/1.1)
    - Código numérico y “en inglés” del estado

Códigos de estado	Significado
1xx	Mensaje informativo
2xx	Resultado exitoso (200 OK)
3xx	Redirección del cliente a otra URL (301 Moved permanently)
4xx	Error en el lado del cliente (404 Not found)
5xx	Error en el lado del servidor (500 Server error)



# Formato de los mensajes

---

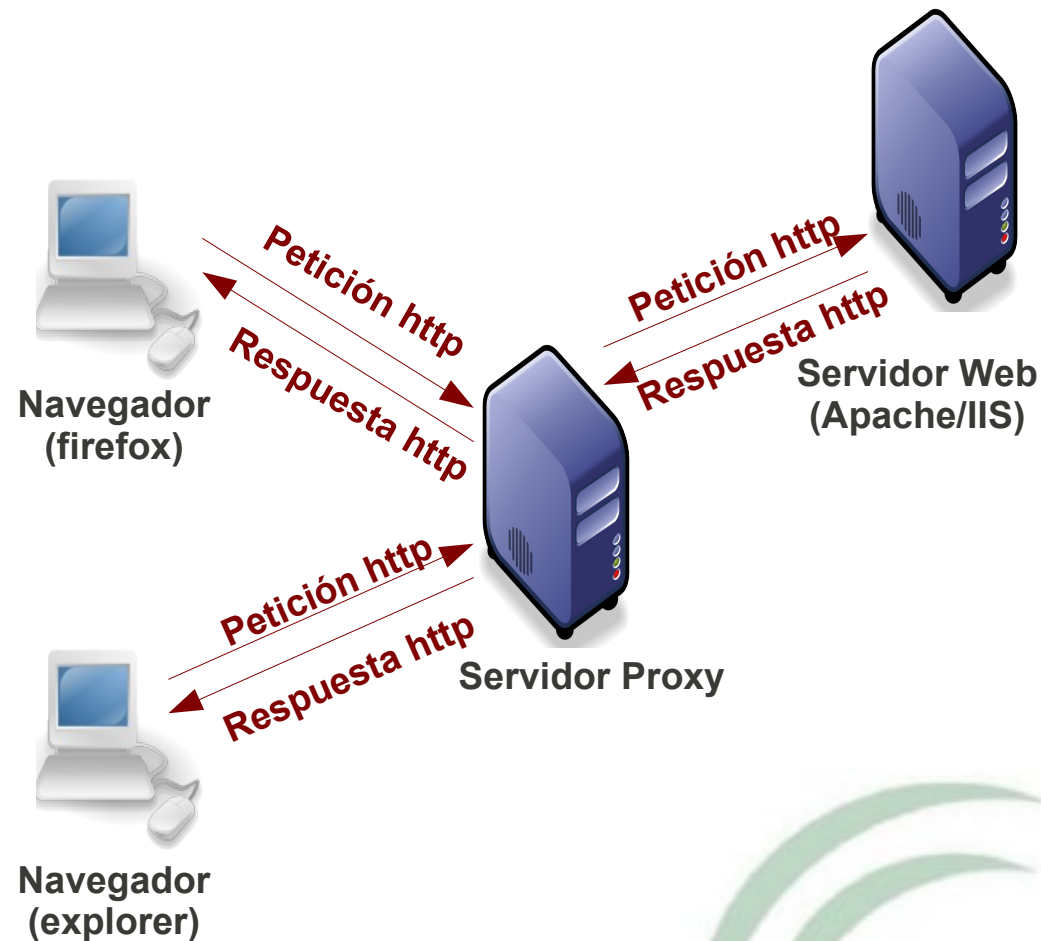
- Líneas de cabecera:
  - En HTTP 1.1 existen 46 cabeceras, siendo la cabecera `Host:` obligatoria en las peticiones
  - Se recomienda incluir en las peticiones al menos:
    - `User-agent:` (ej `Mozilla/4.7`)
  - Se recomienda incluir en las respuestas al menos:
    - `Server:` (ej `Apache/1.3`)
    - `Last-Modified:` (fecha GMT, usada por las caches)





# Representante (proxy) de HTTP

- Un proxy HTTP es un intermediario entre un cliente y un servidor
- Actúa a la vez como cliente y como servidor
- Normalmente un proxy lo es de varios clientes y tiene una caché asociada
- Usos: cortafuegos y aumento de la velocidad por uso de la caché



# HTTPS

---

- HTTP sobre SSL (Secure Socket Layer)
- La conexión TCP está cifrada, de forma que una tercera parte no pueda conocer su contenido
- Permite enviar datos *sensibles* a un servidor web, y recibirlos de él
- Suele utilizar otros mecanismos (como certificados) para ofrecer una mayor seguridad
- Las URL son *https://*
- El puerto estándar es el 443



# Persistencia de estado en HTTP

---

- HTTP se diseña de forma que los servidores no almacenen estado sobre los clientes
- Sin embargo, es muy frecuente la necesidad de mantener un estado persistente entre distintas operaciones de un mismo cliente con un servidor (carro de la compra, login de usuario...)
- Soluciones:
  - Mantener estado de forma externa a HTTP (por IP, por ejemplo)
  - Mediante URLs incluidas en las páginas servidas
  - Mediante campos ocultos
  - Mediante cookies



## Galletitas (cookies)

---

- Normalmente son datos asociados a un usuario (carro de la compra, cuenta de usuario, preferencias...)
- Las galletitas son generadas por los servidores y presentadas por lo clientes en ciertas ocasiones
- Cuando un cliente pide una URL, buscará en su lista de galletitas por si hay alguna que tenga que enviar
- Especificación original de Netscape, luego propuesta como RFC 2109, ampliada en RFC 2965



# Referencias

---

- James Marshall, HTTP Made Really Easy. A Practical Guide to Writing Clients and Servers,  
<http://www.jmarshall.com/easy/http/>
- RFC 1945, HTTP 1.0,  
<http://www.faqs.org/rfcs/rfc1945.html>
- RFC 2068, HTTP 1.1,  
<http://www.faqs.org/rfcs/rfc2068.html>
- RFC 2964, Use of HTTP State Management,  
<http://www.faqs.org/rfcs/rfc2964.html>
- RFC 2965, HTTP State Management Mechanism,  
<http://www.faqs.org/rfcs/rfc2965.html>

