

HTTP

- ✦ URI
- ✦ HTTP 1
- ✦ Proxy web
- ✦ Cookies

HTTP

1

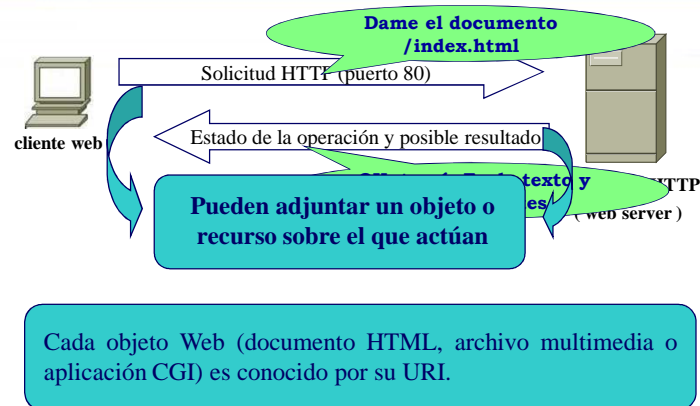
HTTP 1 (*Hypertext Transfer Protocol*)

- ✦ Protocolo basado en texto de recuperación de información
- ✦ Mecanismo *request – response*
- ✦ Originalmente para contenido estático
- ✦ No mantiene estado (no establece una sesión)

HTTP

2

HTTP (*Hypertext Transfer Protocol*)



HTTP

3

HTTP: recursos

- ✦ Un recurso es un bloque de información identificado por su URI (Uniform Resource Identifier)
- ✦ Puede ser un archivo (físico) o generado por un programa (abstracto)
- ✦ URL: Uniform Resource Locator
- ✦ URN: Uniform Resource Name
- ✦ RFC 1630, 2396, 2718, 3305, 3986
- ✦ Internationalized Resource Identifiers (IRIs): RFC 3987

HTTP

4

URI (ver RFC 3986 3.3)

`<scheme>://<authority><path>?<query>`

- ✦ El path termina con el primer "?" o "#" o si no hay más caracteres
- ✦ Puede ser relativo o absoluto
- ✦ Si representa una aplicación puede recibir parámetros

- `http:www.example.org/path/name?param1;param2;param3`
- `/relative_path/name?user=".." ;pwd="..."`
- `../../logo.png`

HTTP

5

URL

- ✦ Identifica un recurso por su ubicación (*location*)

`<scheme>://<user>:<password>@<host>:<port>/<path>?<query>`

HTTP

6

URL: sintaxis

- ✦ Sólo caracteres ASCII «imprimibles»

`<scheme>://<user>:<password>@<host>:<port>/<path>?<query>`

Protocolo a utilizar: por
ejemplo HTTP, FTP, FILE

`http://`

7

Sintaxis URL

`<scheme>://<user>:<password>@<host>:<port>/<path>?<query>`

Algunos sitios o recursos
requieren autenticación

`http://soyyo:miclave@`

8

URL: sintaxis

`<scheme>://<user>:<password>@<host>:<port>/<path>?<query>`

Host: puerto (opcional si es 80)

`http://soyyo:miclave@www.ensitio.com:90`

`http://192.168.0.100`

9

URL: sintaxis

`<scheme>://<user>:<password>@<host>:<port>/<path>?<query>`

Identificación del recurso
dentro del servidor

`http://soyyo:miclave@www.ensitio.com:90/index.html`

10

URL: sintaxis

`<scheme>://<user>:<password>@<host>:<port>/<path>?<query>`

Parámetros

`http://soyyo:miclave@www.ensitio.com:90/pagZZ.html?width=1024&lang=es`

11

URL: sintaxis

✦ Puede incluir al final un "fragmento"

`http://www.ensitio.com/intro.html#chapter1`

12

URN

- ✦ Identifica un recurso por su nombre
- ✦ No implica que el recurso exista o cómo acceder a él

`urn:isbn:0132856204`

`urn:isan:0000-0002-3C36-0000-Y-0000-0000-9`

`urn:uuid:6e8bc430-9c3a-11d9-9669-0800200c9a66`

`urn:www.apache.org:`

`http://www.iana.org/assignments/urn-namespaces/urn-namespaces.xhtml`

URI Scheme

- ✦ URIs pueden ser usadas para acceder a recursos, ya sea por medio de URL o URN
- ✦ Ejemplo en html

```
<a href="/img/logo.png">  
<a href="urn:isbn:0453457513">
```

Mensajes HTTP

- ✦ El cliente y el servidor intercambian mensajes HTTP
 - ✦ *Request message* (cliente → servidor)
 - ✦ *Response message* (servidor → cliente)



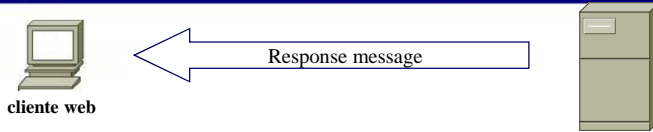
- ✦ Solicitudes (*Request*)
 - ✦ **GET**: Solicita un recurso al servidor
 - ✦ **HEAD**: Solicita solo los headers del recurso.
 - ✦ **POST**: Envía información al servidor para ser procesada

Mensajes HTTP



- ✦ Solicitudes (*Request*) menos utilizados
 - ✦ **PUT**: Envía un recurso al servidor
 - ✦ **TRACE**: Analiza el recorrido de la solicitud.
 - ✦ **OPTIONS**: Consulta los métodos disponibles en el servidor.
 - ✦ **DELETE**: Elimina un recurso del servidor.

Mensajes HTTP



★ Respuestas (*Response*)

★ **Start Line:** Versión, código de respuesta y mensaje.

1XX	Información – El proceso continúa
2XX	Éxito - Acción recibida, comprendida y aceptada
3XX	Redirección – Se requiere nueva acción
4XX	Error en cliente – Sintaxis y/o Semántica inválida
5XX	Error en servidor – Falló pedido correcto

17

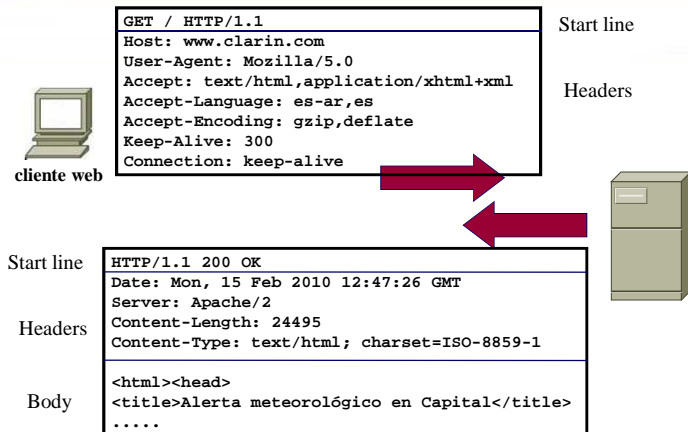
Tipos de información

- ★ Utiliza MIME para describir contenido multimedia
- ★ Cada objeto es etiquetado por el web server
 - ★ text/html
 - ★ text/plain
 - ★ video/mp4
 - ★ image/jpeg
 - ★ application/pdf
 - ★ ...

HTTP

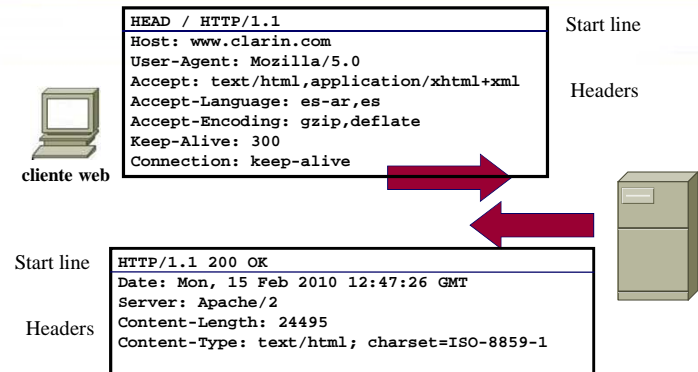
18

Ejemplo: HTTP - GET



19

Ejemplo: HTTP - HEAD



20

Ejemplo: HTTP - POST



cliente web

```
POST /itbaV/mynav.asp HTTP/1.1
Host: iol.itba.edu.ar
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml+xml
Referer: http://iol.itba.edu.ar/itbaV/welcome.asp
Content-Type: application/x-www-form-urlencoded
Content-Length: 76
```

Body

```
secretNr=378924198&txtdni=12345678&txtpwd=mipass&Submit=Conectar&cmd=login
```

```
HTTP/1.1 200 OK
Connection: close
Date: Mon, 15 Feb 2010 17:05:22 GMT
Server: Microsoft-IIS/6.0
Content-Length: 11276
Content-Type: text/html
```

```
<html><head>
<title>ITBA OnLine</title>
.....
```

21

Ejemplo: HTTP - GET



cliente web

```
GET /itbaV/mynav.asp?secretNr=378924198&txtdni=
12345678&txtpwd=mipass&Submit=Conectar&cmd=login
```

```
Host: iol.itba.edu.ar
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml+xml
Referer: http://iol.itba.edu.ar/itbaV/welcome.asp
Content-Type: application/x-www-form-urlencoded
Content-Length: 76
```

22

Get vs Post

- ★ GET requests
 - ★ pueden ser "cacheados"
 - ★ el browser los mantiene en el historial
 - ★ pueden ser "bookmarked"
 - ★ tienen longitud acotada
 - ★ sólo para pedir datos
- ★ POST requests
 - ★ nunca son "cacheados"
 - ★ no se mantienen en el historial del browser
 - ★ no pueden ser "bookmarked"
 - ★ no tienen restricción de longitud de datos

Ejemplo: HTTP - OPTIONS



cliente web

```
OPTIONS * HTTP/1.1
User-Agent: Mozilla/4.0 (compatible;
MSIE5.01; Windows NT)
```

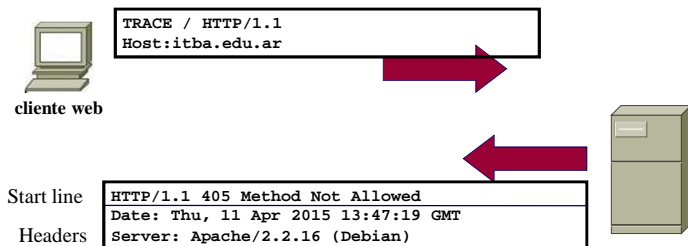
Start line

Headers

```
HTTP/1.1 200 OK
Date: Mon, 01 Aug 2016 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Allow: GET,HEAD,POST,OPTIONS,TRACE
Content-Type: httpd/unix-directory
```

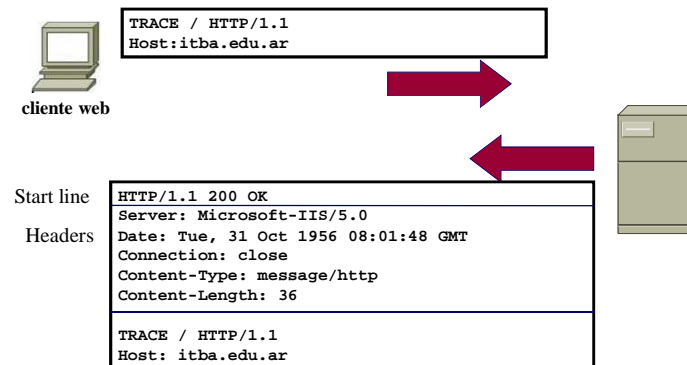
24

Ejemplo: HTTP - TRACE



25

Ejemplo: HTTP - TRACE



26

Headers HTTP

- ✦ Estructura: <nombre>: <valor asociado>
 - ✦ Finalizan con una línea en blanco.
 - ✦ Tipos:
 - ✦ Generales
 - ✦ De solicitudes
 - ✦ De respuestas
 - ✦ De contenido
- Usualmente se envían en este orden

27

Headers generales

- ✦ Cache-control: Directivas para cache
- ✦ Connection: Se definen opciones de conexión
- ✦ Date: Fecha de creación del mensaje
- ✦ Transfer-Encoding: Indica el encoding de transferencia
- ✦ Via: Muestra la lista de intermediarios por los que pasó el mensaje.

28

Headers de solicitud

- ✦ Accept : Tipo de contenido aceptado por el cliente
- ✦ Accept-Charset: Charset (ISO-xxxx, UTF-8) aceptado por el cliente
- ✦ Accept-Encoding: Encoding (gzip, compress) aceptado por el cliente
- ✦ Expect: Comportamiento esperado del server frente al request.
- ✦ From: Email del usuario de la aplicación que generó el request.
- ✦ Host: Servidor y puerto destino del request.
- ✦ If-Modified-Since: Condiciona al request a la fecha indicada.
- ✦ Referer: URL del documento que generó el request
- ✦ User-Agent: Aplicación que generó el request
- ✦ Upgrade: solicita que use otro protocolo
- ✦ Range: solicita un rango (en bytes) del recurso

29

Headers de respuesta

- ✦ Age: Estimación en segundos del tiempo que fue generada la respuesta en el server.
- ✦ Connection:
- ✦ Location: URI a redireccionar
- ✦ Retry-After: Tiempo de delay para reintento
- ✦ Server: Descripción del software del server.
- ✦ Authorization: indica que el recurso necesita autorización

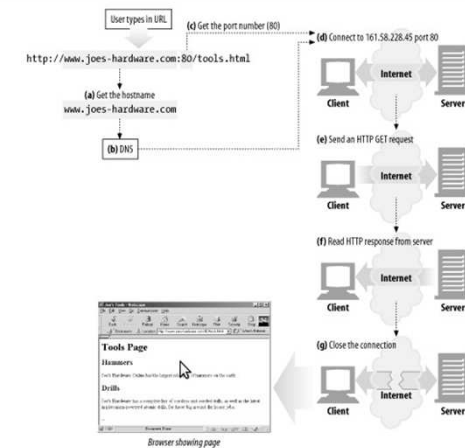
30

Headers de contenido

- ✦ Allow: Métodos aplicables al recurso.
- ✦ Content-Encoding
- ✦ Content-Length
- ✦ Content-Location
- ✦ Content-MD5
- ✦ Content-Type
- ✦ Expires: Fecha de expiración del recurso.
- ✦ Last-Modified: Fecha de modificación del recurso

31

Proceso de conexión



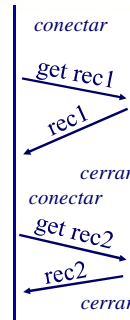
32

Conexiones HTTP

- ✦ Conexión no persistente
 - ✦ Se establece conexión TCP
 - ✦ Se solicita el recurso
 - ✦ Se cierra la conexión TCP
- ✦ Conexión persistente
 - ✦ Se establece la conexión
 - ✦ Se solicita recurso 1
 - ✦ Se solicita recurso 2
 - ✦ ...
 - ✦ Se cierra la conexión TCP

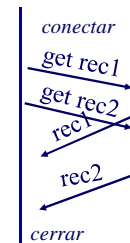
Conexiones HTTP

Origen Destino



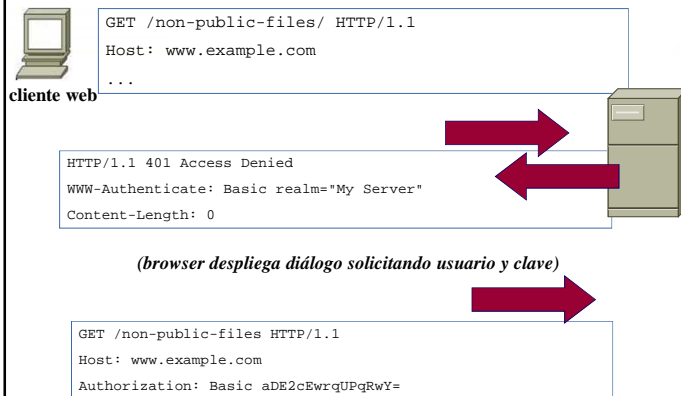
No persistente

Origen Destino



Persistente

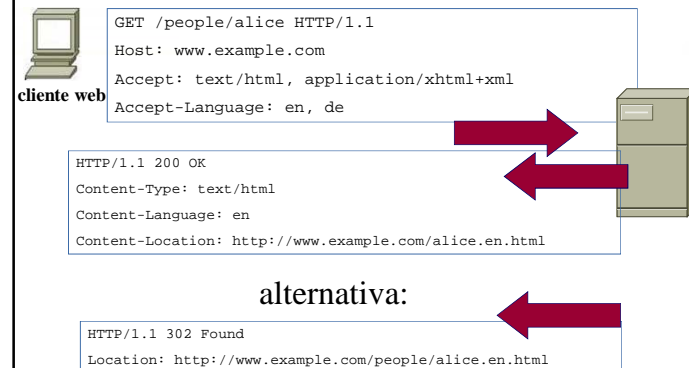
Autorización



HTTP

35

Negociación de contenido



HTTP

36

HTTP "a mano"

```
user@server:~$ telnet itba.edu.ar 80
Trying 190.104.250.104...
Connected to itba.edu.ar.
Escape character is '^]'.
GET /es/area/grado HTTP/1.1
Host: itba.edu.ar
```

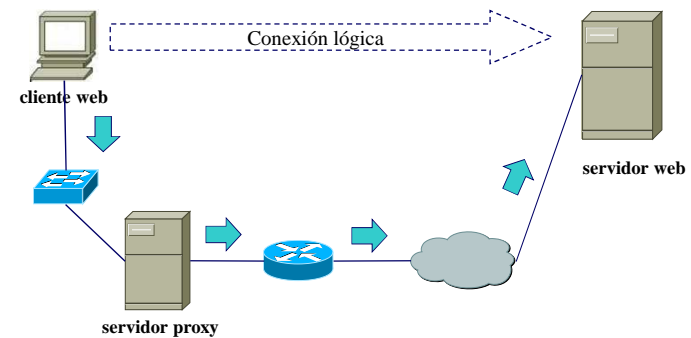
(código fuente del recurso)

Usar Wireshark para "sniffear" el contenido completo de los paquetes intercambiados

ver nc, curl

Proxy servers

Un *proxy server* actúa como intermediario entre una aplicación cliente y un web server.

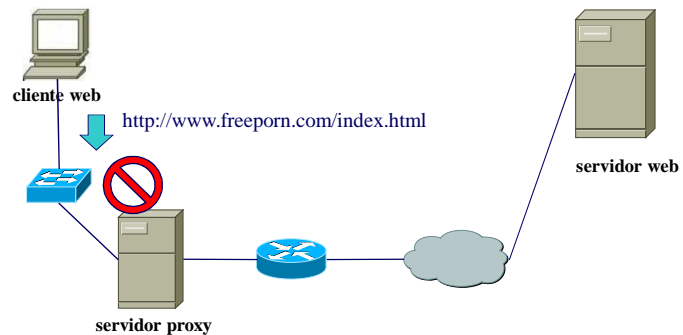


HTTP

38

Proxy servers

Un *proxy server* permite controlar el acceso a determinados sitios o páginas, y almacenar en caché recientes consultas.

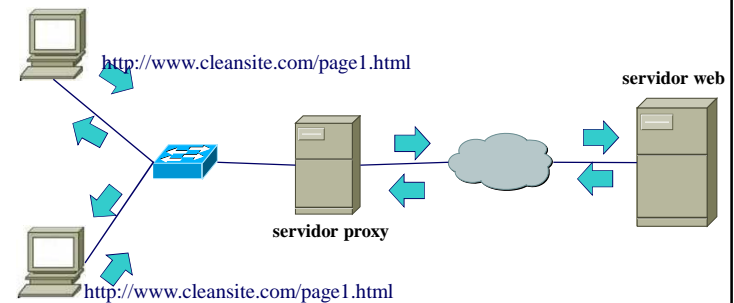


HTTP

39

Proxy servers

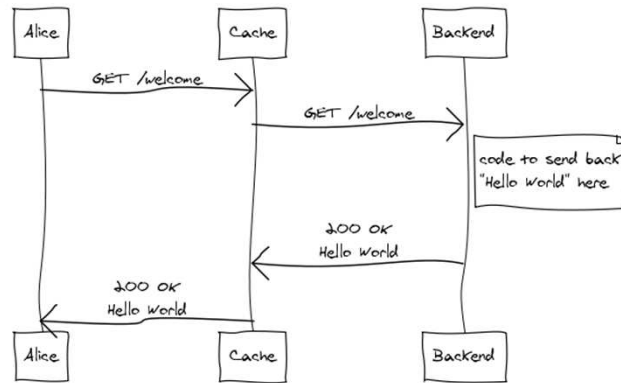
Un *proxy server* permite controlar el acceso a determinados sitios o páginas, y almacenar en caché recientes consultas.



HTTP

40

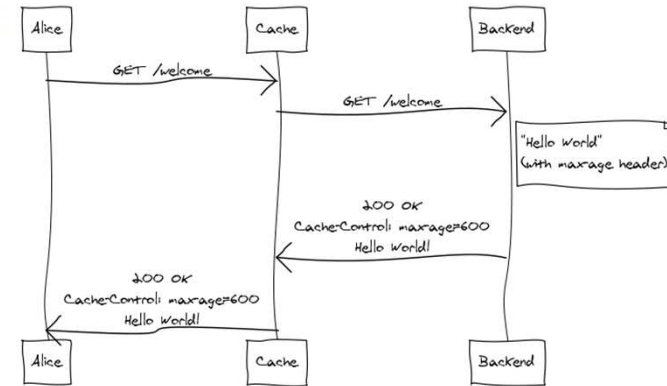
Cache: sin directivas



HTTP

41

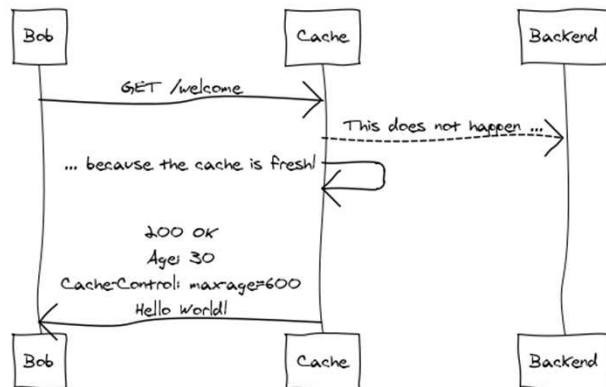
Cache: expiración 1/2



HTTP

42

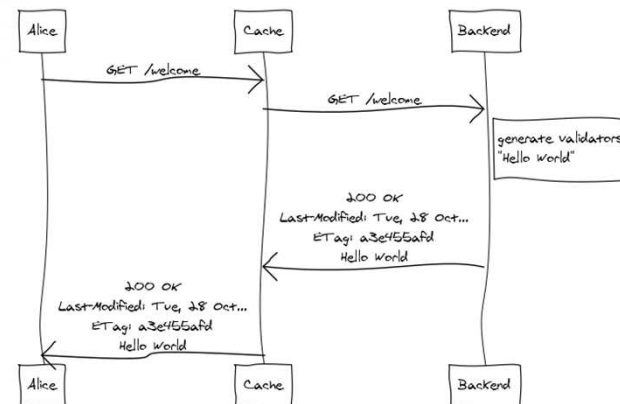
Cache: expiración 2/2



HTTP

43

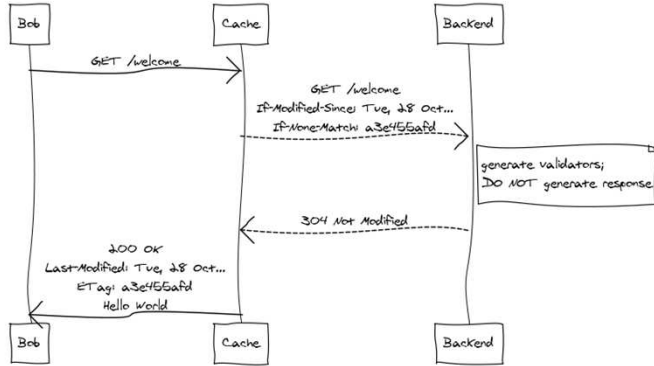
Cache: validación 1/2



HTTP

44

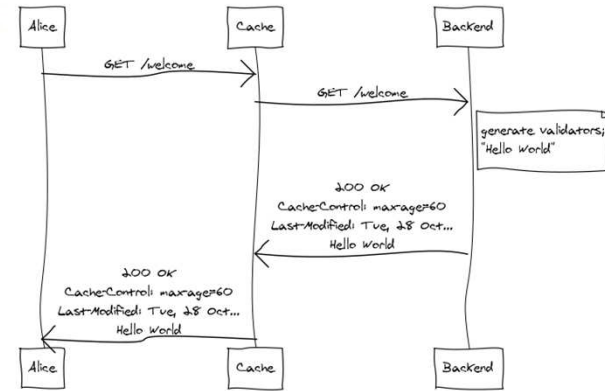
Cache: validación 2/2



HTTP

45

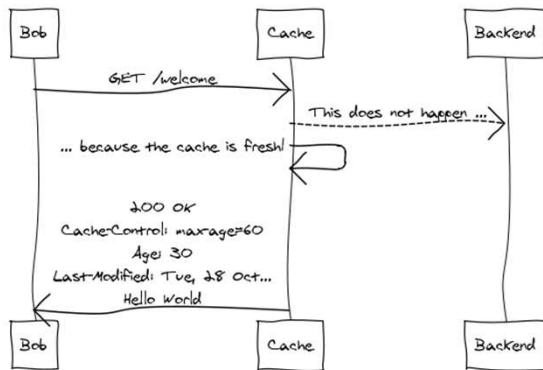
Combinando ambas 1/3



HTTP

46

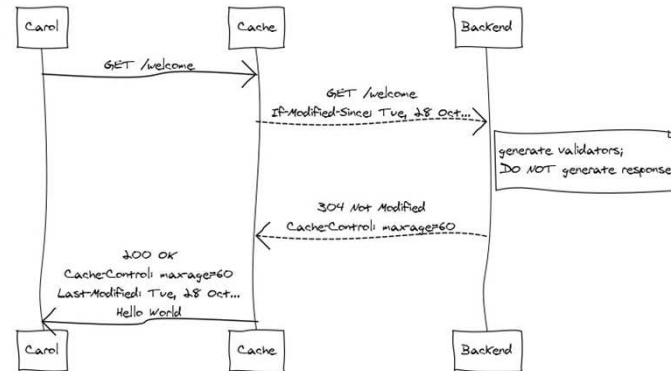
Combinando ambas 2/3



HTTP

47

Combinando ambas 3/3



HTTP

48

Proxy y HTTP TRACE



```
TRACE / HTTP/1.1
Host: www.example.com
```

```
HTTP/1.1 200 OK
Date: Sat, 16 Mar 2013 22:05:22 GMT
Server: Apache
Content-Type: message/http

X-Secret-info:
lkjfklsfoij2oif4oijalskdfjsecretstringoeshe
re12342134

Obfuscated-Client-IP: 123.123.123.123
Content-Length: 348
```

49

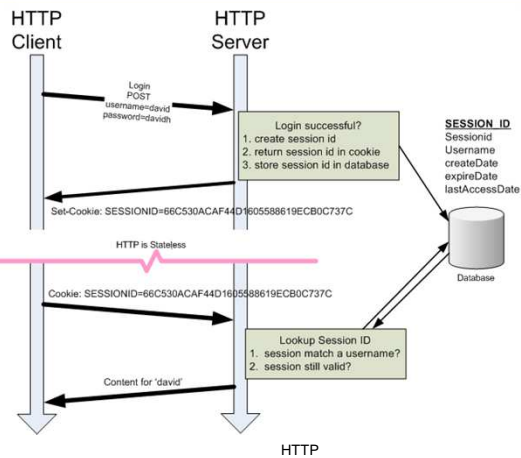
Cookies

- ✦ Pequeña pieza de información enviada por el servidor y almacenada por el browser
- ✦ Usadas para mantener un estado entre el cliente y el servidor
 - ✦ Servidor envía cookie a cliente HTTP («*set-cookie*» *response header*)
 - ✦ Cliente HTTP retorna cookie al servidor («*cookie*» *request header*)
- ✦ Persistencia
 - ✦ Session cookie
 - ✦ Persistent cookie
- ✦ Third-party cookie
- ✦ Secure cookie

HTTP

50

Cookies



51

Cookies



```
GET /index.html HTTP/1.1
Host: www.example.com
...
```

```
HTTP/1.1 200 OK
Content-Type: text/html
Set-Cookie: sessionId=123xyz; Expires=Wed, 04 May 2016 06:01:00 GMT
Set-Cookie: lastSession=2016/03/01
...
```

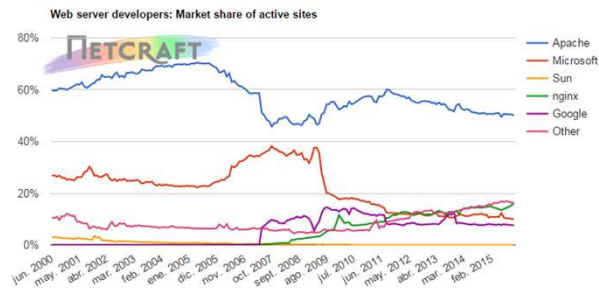
```
GET /menu.html HTTP/1.1
Host: www.example.com
Cookie: sessionId=123xyz; lastSession=2016/03/01
...
```

HTTP

52

Web server

Estadística de servidores web (www.netcraft.com)



HTTP

53

Web semántica

Un buscador convencional

- busca coincidencia de palabras y frases
- prioriza la concordancia de acuerdo al "peso" dentro del documento

Buscador Actual

Resultados de la búsqueda:

[Toda la magia de Budapest y Praga](#)
... Suplementos Gran Premio Fórmula 1 en Budapest **para** las salidas del ... con Ferias y/o Congresos en **Praga** del 9 ... Más información de los **vuelos** ...

[LA VANGUARDIA DIGITAL - Praga, testigo de la historia europea](#)
... Para emergencias el teléfono de la policía es el 150, el de las ambulancias el ... 46) y **Praga** tres días **por** semana. Los **vuelos** salen de Madrid (Tel ...

[Foros sobre Europa República Checa Praga inquietante](#)
... solo decirte que me llamó la atención tu alias (aunque no me llamo Raula) y que me voy **mañana** mismo **para** **Praga** ... buscador de **vuelos** ...

[ofertas de espectáculos, viajes y hoteles al mejor precio](#)
... autoridades que tienen tres copas gigantes **para** entregar a ... **mañana** creo que cogeremos el bus **mañana** ... En Atrápalo puedes también reservar **vuelos** ...

Web semántica

Un buscador semántico

- detecta desde dónde el usuario envía la pregunta
- el primer "mañana" es el de día hoy + 1
- el segundo "mañana" representa un lapso del día

Buscador Semántico

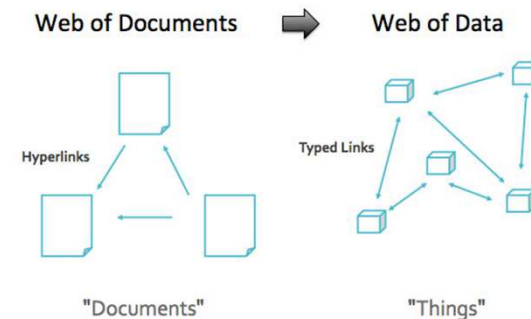
Resultados de la búsqueda:

[viajaconnosotros.com - viajes a Praga](#)
... todos los **vuelos** a **Praga** desde tu ciudad que saldrán **mañana por la mañana**, ordenados según su hora de salida

[viajes a Praga - vuelos disponibles](#)
... lista de **vuelos**. Horarios de salida y llegada ...

[Ofertas especiales - vuelos a Praga](#)
... ofertas especiales de **vuelos** a **Praga** ...

URIs: Web of Documents vs Web of data



HTTP

56

Linked Data

- ✦ Es una forma de vincular los distintos datos que están distribuidos en la Web
 - ✦ Usar URIs para identificar las cosas
 - ✦ Usar URIs HTTP
 - ✦ Ofrecer información sobre los recursos usando RDF
 - ✦ Incluir enlaces a otros URIs

[https://www.w3.org/standards/semanticweb/
data](https://www.w3.org/standards/semanticweb/data)