

# Programación WEB (PWEB)

## Tema 1.3

### Protocolo HTTP

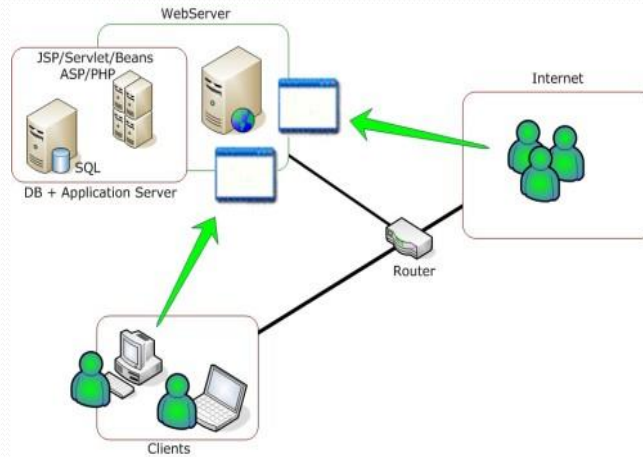


Ramón Alcarria  
Miguel Ángel Manso  
Emerson Casteñeda<sub>1</sub>

## Contenido

- Arquitectura de Internet
- Protocolos de Internet
- HTTP
  - URLs
  - Methods
  - Headers
  - Códigos de estado
- Herramientas para realizar peticiones HTTP

# Arquitectura de Internet

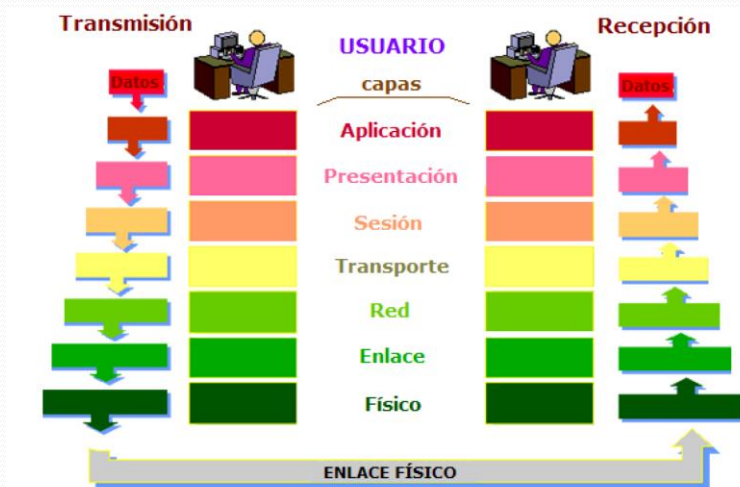


PWEB 2014-2015

3

# Arquitectura de Internet

- Heredada del modelo OSI



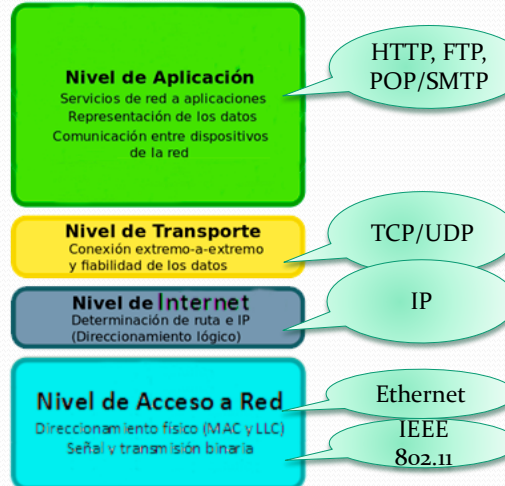
4

# Arquitectura de Internet

## LA PILA OSI



## LA PILA TCP/IP



PWEB 2014-2015

5

# Protocolos de Internet

- *De navegación Web*
  - HTTP (*Hypertext transport protocol*)
    - ❖ *Define operaciones sobre dominios*
    - E.g. <http://www.whitrieia.ac.nz>**
  - Otros estándares:
    - URL (*Universal Resource Locator*)
      - ❖ *Define recursos en Internet.*
      - [www.whitrieia.ac.nz](http://www.whitrieia.ac.nz)**
    - HTML (*Hypertext Mark-up Language*)
      - ❖ *Lenguaje de marcado para páginas Web*



PWEB 2014-2015

6

# Protocolos de Internet

- *De email*
  - STMP (*Simple Mail Transfer Protocol*)
    - ❖ Permite el envío de mensajes de email entre servidores.
    - ❖ Los emails se envían mediante STMP
  - POP (*Post Office Protocol*)
    - ❖ Utilizado por los clientes de e-mail para recuperar correos de un servidor remoto
    - ❖ Los emails se reciben mediante POP



# HTTP

- *HyperText Transport Protocol*
- Lenguaje de la Web
  - Protocolo usado para comunicar clientes y servidores
- Protocolo orientado a conexión (TCP) y el puerto por defecto es (80)
- RFC 1945

# HTTP

- **Conexión.-** establecimiento de una conexión cliente-servidor. Usualmente puerto por defecto (80) aunque pueden usarse otros no reservados
- **Solicitud.-** envío por parte del cliente de un mensaje de solicitud al servidor (GET, POST, HEAD, ..)
- **Respuesta.-** envío por parte del servidor de una respuesta al cliente
- **Cierre.-** fin de la conexión por parte del cliente y el servidor

HTTP es un protocolo sin estado es decir que no guarda ninguna información sobre conexiones anteriores

# HTTP - URLs

- URL
  - *Uniform Resource Locator*
    - **protocol** (http, ftp, news)
    - **host name** (name.domain name)
    - **port** (usually 80)
    - directory **path** to the resource
    - **resource** name
  - **http://www.myplace.com/www/index.html**
  - **http://www.myplace.com:80/cgi-bin/t.exe**

# HTTP - methods

- Los métodos que ofrece el protocolo son: Get, Post, Head, Put, Delete, Options, Trace, Connect
  - **GET** (<http://xx/pp/serv?par1=val1&par2=val2>)
    - Recupera un recurso de un servidor
      - Petición de página web sencilla
      - Ejecutar un programa CGI
      - Ejecutar un CGI con **parámetros en la URL**
  - **POST**
    - Método preferido para procesar formularios
    - También para ejecutar programas CGI
    - Los datos se incluirán en el cuerpo de la petición
    - Más seguro y privado

# HTTP - methods

- Como hacer un GET:

```

1 <form method="GET" action="foo.php">
2
3 First Name: <input type="text" name="first_name" /> <br />
4 Last Name: <input type="text" name="last_name" /> <br />
5
6 <input type="submit" name="action" value="Submit" />
7
8 </form>

```

Al aceptar el form se crea el siguiente mensaje:

```

1 GET /foo.php?first_name=John&last_name=Doe&action=Submit HTTP/1.1
2 ...

```

# HTTP - methods

- Como hacer un POST:

```

1 <form method="POST" action="foo.php">
2
3 First Name: <input type="text" name="first_name" /> <br />
4 Last Name: <input type="text" name="last_name" /> <br />
5
6 <input type="submit" name="action" value="Submit" />
7
8 </form>

```

Al aceptar el form se crea el siguiente mensaje:

```

01 POST /foo.php HTTP/1.1
02 Host: localhost
03 User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5)
04 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
05 Accept-Language: en-us,en;q=0.5
06 Accept-Encoding: gzip,deflate
07 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
08 Keep-Alive: 300
09 Connection: keep-alive
10 Referer: http://localhost/test.php
11 Content-Type: application/x-www-form-urlencoded
12 Content-Length: 43
13
14 first_name=John&last_name=Doe&action=Submit

```

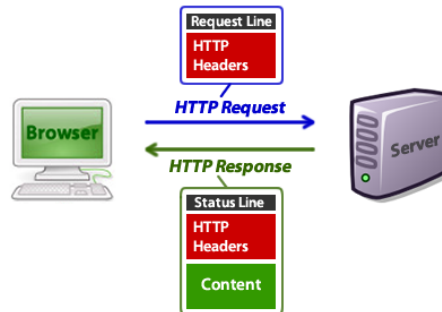
13

# HTTP - methods

- HEAD
  - *"When you send a HEAD request, it means that you are only interested in the response code and the HTTP headers, not the document itself."*
  - Se utiliza para gestionar las URL condicionales y poder manejar esquemas que mejoran el funcionamiento (*used for conditional URL handling for performance enhancement schemes*)
    - Obtiene el recurso de la URL si no existe una copia local o si ésta es muy antigua respecto a los datos del servidor

# HTTP Headers

- Cabeceras del protocolo HTTP
- Contienen información sobre el navegador cliente, la página pedida, el servidor, etc.



PWEB 2014-2015

15

# HTTP Headers

- Preceden a las peticiones HTTP
- <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>
- Campos de la cabecera:

- |                   |                     |
|-------------------|---------------------|
| - From            | - Accept            |
| - Accept-Encoding | - Accept Language   |
| - Referer         | - Authorization     |
| - Charge-To       | - If-Modified-Since |
| - Pragma          |                     |

PWEB 2014-2015

16



# HTTP Headers

Petición (Request)

| method | path  | protocol |
|--------|---|----------|
| GET    | /tutorials/other/top-20-mysql-best-practices/ | HTTP/1.1 |

```

Host: net.tutsplus.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120
Pragma: no-cache
Cache-Control: no-cache
  
```

**HTTP headers as Name: Value**

PWEB 2014-2015

17

# HTTP Headers

Respuesta (Response)

```

HTTP/1.x 200 OK
Transfer-Encoding: chunked
Date: Sat, 28 Nov 2009 04:36:25 GMT
Server: LiteSpeed
Connection: close
X-Powered-By: W3 Total Cache/0.8
Pragma: public
Expires: Sat, 28 Nov 2009 05:36:25 GMT
Etag: "pub1259380237.gz"
Cache-Control: max-age=3600, public
Content-Type: text/html; charset=UTF-8
Last-Modified: Sat, 28 Nov 2009 03:50:37 GMT
X-Pingback: http://net.tutsplus.com/xmlrpc.php
Content-Encoding: gzip
Vary: Accept-Encoding, Cookie, User-Agent

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Top 20+ MySQL Best Practices - Nettuts+</title>
<!-- ... rest of the html ... -->
  
```

PWEB 2014-2015

18

# HTTP Headers

- Problema del GET:
- Límite en el tamaño de las cabeceras.
- HTTP does not define any limit. However most web servers do limit size of headers they accept.
- For example in Apache default limit is 8KB, in IIS it's 16K.
- Server will return 413 Entity Too Large error if headers size exceeds that limit.

# HTTP Headers

## ***From:***

- Debe contener una dirección de e-mail correspondiente con el usuario que ha realizado la petición.
- Ejemplo:
  - From: webmaster@w3.org

## ***Accept:***

- Lista de esquemas que acepta el cliente
  - Accept: text/html
  - Accept: audio/basic q=1
  - if no Accept is found; plain/text is assumed

# HTTP Headers

## ***Accept-Language:***

- Idioma configurado por el usuario
  - Accept-Language: en-us,en;q=0.5
- Puedes indicar varios idiomas separados por comas

## ***Accept-Encoding:***

- Similar a Accept, para las codificaciones
- Ejemplo:
  - Accept-Encoding: x-compress;x-zip
  - Accept-Encoding: gzip,deflate

# HTTP Headers

## ***User-Agent:***

- Información del cliente:
  - Nombre y version del navegador
  - Sistema operativo y version
  - Idioma por defecto
- Ejemplos:
  - User-Agent: IBM WebExplorer DLL /v960311
  - Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.13 (KHTML, like Gecko) Chrome/o.2.149.29 Safari/525.13
  - Mozilla/5.0 (Linux; U; Android 3.0; en-us; Xoom Build/HRI39) AppleWebKit/534.13 (KHTML, like Gecko) Version/4.0 Safari/534.13

# HTTP Headers

## **Referer:**

- Para que los servidores sepan de qué página procede la petición URL
- Esto permite que los servidores realicen auditoría en relación al lugar del que han obtenido el link de acceso actual, poder analizar enlaces rotos, etc.
- Ejemplo
  - Referer: `http://www.w3.com/xxx.html`

# HTTP Headers

## **Authorization:**

- Para esquemas de autenticación con contraseñas
- Ejemplos:
  - Authorization: user fred:mypassword
  - Authorization: kerberos kerberosparameters

## **Charge-To:**

- Información relativa a quien se ha de realizar el cargo (cobro) por una operación
- Depende del sistema de cargos/cobros

# HTTP Headers

## ***Pragma:***

- Equivalente en formato a *accept*
- Destinado para servidores
- Los proxy deben de ser transparentes, aunque lo usan
- Solo el *pragma currently* indica que no es cacheable; los proxy deben obtener el recurso desde el servidor y no desde la cache

# HTTP Headers

## ***If-Modified-Since:***

- Usado con el método GET para realizar peticiones condicionadas a una fecha de modificación
- Si el recurso no ha sido modificado desde la fecha especificada, el servidor retorna una cabecera con el código 304 “Not modified” en vez de retornar el recurso
  - En éstos casos el cliente puede mostrar la versión cacheada
  - If-Modified-Since: Sat, 28 Nov 2009 06:38:19 GMT

## Códigos de estado

- Código que retorna el servidor
- [http://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](http://en.wikipedia.org/wiki/List_of_HTTP_status_codes)
- Códigos:
  - 1xx - reserved for future use
  - 2xx - successful, understood and accepted
  - 3xx - further action needed to complete
  - 4xx - bad syntax in client request
  - 5xx - server can't fulfill good request

## Códigos de estado

- 100 Continua
- 101 Cambio de protocolo
- 200 OK
- 201 Creado
- 202 Aceptado
- 203 Información no oficial
- 204 Sin Contenido
- 205 Contenido para reset
- 206 Contenido parcial
- 300 Múltiples posibilidades
- 301 Mudado permanentemente
- 302 Encontrado
- 303 Vea otros
- 304 No modificado
- 305 Utilice un proxy
- 307 Redirección temporal
- 400 Solicitud incorrecta
- 401 No autorizado
- 402 Pago requerido
- 403 Prohibido
- 404 No encontrado
- 405 Método no permitido
- 406 No aceptable
- 407 Proxy requerido
- 408 Tiempo de espera agotado
- 409 Conflicto
- 410 No mapas disponible
- 411 Requiere longitud
- 412 Falló precondition
- 413 Entidad de solicitud demasiado larga
- 414 URI de solicitud demasiado largo
- 415 Tipo de medio no soportado
- 416 Rango solicitado no disponible
- 417 Falló expectativa
- 500 Error interno
- 501 No implementado
- 502 Pasarela incorrecta
- 503 Servicio no disponible
- 504 Tiempo de espera de la pasarela agotado
- 505 Versión de HTTP no soportada

# Códigos de estado

The image displays four browser windows illustrating different HTTP status codes:

- 404 Not Found:** The requested URL /blah was not found on this server.
- 401 Authorization Required:** This server could not verify that you are authorized to access the document requested. Either you supplied the wrong credentials (e.g., bad password), or your browser doesn't understand how to supply the credentials required.
- 403 Forbidden:** You don't have permission to access /images/ on this server.
- 500 Internal Server Error:** The server encountered an internal error or misconfiguration and was unable to complete your request. Please contact the server administrator, admin@localhost and inform them of the time the error occurred, and anything you might have done that may have caused the error.

PWEB 2014-2015 29

# Herramientas

- Extension *http resource test* para Firefox
  - (<https://addons.mozilla.org/es-es/firefox/addon/http-resource-test/>)

The image shows the Firefox browser interface with the 'Herramientas' (Tools) menu open, highlighting the 'HTTP Resource Test' extension. The extension's window is also visible, displaying the following details:

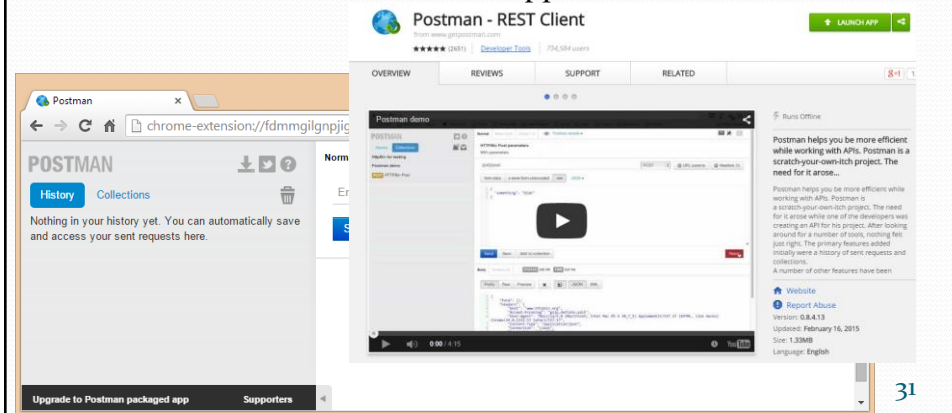
- Resource:** http://www.upm.es
- Method:** GET
- Client Request:** Headers
- Server Response:**
  - HTTP/1.1 200 OK
  - Content-Length: 4530
  - Content-Encoding: gzip
  - Date: Thu, 16 Feb 2012 22:33:50 GMT
  - Set-Cookie: JSESSIONID=A03FBD2D599C96977BE4FACB33AE4856; Path=/
- Body:**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html xmlns="http://www.w3.org/1999/xhtml" lang="es-ES" > <head> <title>Universidad Politécnica de Madrid</title> <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /> <meta names="Language" content="ES" /> <meta names="Description" content="Universidad Politécnica de Madrid" /> <meta names="copyright"
```

PWEB 2014-2015

# Herramientas

- Extension **Postman REST Client** para Chrome
  - <http://www.getpostman.com/>
  - Para arrancarlo: `chrome://apps`



# Algunas direcciones

- Buscamos por “public REST API”
  - <http://www.publicapis.com/> (Pide registro)
  - <http://www.predic8.com/rest-demo.htm>
- Por ejemplo:
  - <http://www.thomas-bayer.com/sqlrest/CUSTOMER/>
  - <http://www.thomas-bayer.com/sqlrest/INVOICE/>
  - <http://www.thomas-bayer.com/sqlrest/ITEM/>
  - <http://www.thomas-bayer.com/sqlrest/PRODUCT/>
- <http://httpbin.org/>
  - <http://httpbin.org/post>
  - <http://httpbin.org/put>
  - <http://httpbin.org/delete>



## Algunas direcciones

- Herramientas tipo Chrome Postman
- <https://www.hurl.it/>