

Tema 5. EL PROTOCOL HTTP

En aquest tema veurem com PHP processa i gestiona la informació que el client (el navegador) envia mitjançant l'ús de formularis, al servidor i com aquest genera una resposta segons la petició sol·licitada.

5.1. EL PROTOCOL HTTP

Encara que PHP pot funcionar des de la línia de comandes, el seu ús principal està relacionat amb els llocs Web; de fet, PHP pot ser definit com un servei complementari als proporcionats pels servidors Web. Aquests servidors basen el seu funcionament en l'ús del protocol HTTP (*Protocol de Transferència d'Hipertextos*). Per això, per entendre el funcionament de les tècniques que s'expliquen en aquest tema, es fa necessari un coneixement previ del protocol HTTP.

És un senzill protocol client-servidor que facilita els intercanvis d'informació entre els clients i els servidors mitjançant operacions simples de tipus sol·licitud/resposta. Bàsicament controla la manera en que els clients web sol·liciten recursos dels servidors web i la forma en que aquests els hi envien el recursos de tornada. Tots els recursos proporcionats per un servidor web (documents HTML, gràfics, vídeos, fitxers de so...) estan associats a una URL o *Localitzador Uniforme de Recursos*.

5.1.1. ESTRUCTURA DELS MISSATGES HTTP

Les peticions i respostes s'envien com a missatges de text que es componen de dos parts, una capçalera i un cos (els dos elements es separen al missatge mitjançant una línia en blanc). Només existeixen 2 tipus de missatges:

- un per a realitzar les peticions
- un altre per a la respostes.

Les peticions sempre tenen capçalera i, en algunes ocasions, cos; no obstant, les respostes la majoria de les ocasions tenen els dos components.

L'estructura d'aquests missatges es poden resumir a la següent taula:

Tipus de missatge	Components
petició	Comanda HTTP
	Capçaleres de la petició
	Línia en blanc (separador)
	Informació addicional
resposta	Resultat de la petició
	Capçaleres de la resposta
	Línia en blanc (separador)
	Informació addicional

5.1.1.1. COMANDES HTTP

La primera línia d'un missatge de petició, la comanda HTTP, té la següent estructura:

mètode_HTTP URL_rekurs versió_del_protocol

Proporciona la comanda HTTP utilitzada, la URL del recurs del servidor al qual es vol accedir i la versió del HTTP utilitzada.

A la següent taula es poden veure els mètodes HTTP disponibles:

Mètode	Utilització
Get	Obtenció de recursos del servidor
Post	Enviament d'informació al servidor
Head	Obtenció d'informació sobre un recurs específic del servidor
Put	Actualització d'un recurs al servidor
Delete	Eliminació d'un recurs al servidor
Trace	Obtenció d'informació de diagnòstic sobre la comunicació establerta amb el servidor
Connect	Reservat per a la comunicació mitjançant un <i>proxy</i>
Options	Obtenció d'informació sobre les opcions de comunicació d'un recurs
Link	Creació d'un enllaç entre recursos
Unlink	Eliminació d'un enllaç entre recursos

Els mètodes més utilitzats són *get*, *post* i *head*, a més de ser els més estàndards.

5.1.1.2. CAPÇALERES DE PETICIÓ I RESPOSTA

Les capçaleres de petició proporcionen informació sobre el client Web utilitzat, els tipus de continguts que és capaç d'acceptar, la longitud dels continguts, etc.... Per la seva part, les capçaleres de resposta proporcionen informació al software del servidor i a la naturalesa de la informació inclosa en el cos de la resposta.

El format general d'una capçalera és la següent:

variable_de_capçalera: valor

Existeixen capçaleres comuns a peticions i resultats i capçaleres específiques.

Les capçaleres més habituals són:

Capçaleres comuns	
capçalera	utilització
Content-Type	Tipus MIME de la informació continguda al missatge
Content-Length	Tamany en <i>bytes</i> de la informació continguda al missatge
Content-Encoding	Format de codificació de les dades enviades al missatge
Transfer-Encoding	Tipus de codificació utilitzada en el cos del missatge
Date	Data i hora local de la operació incloent la zona horària
Pragma	Opcions de comportament del protocol HTTP
Cache-Control	Directives de control de cau
Connection	Informació de l'estat de la connexió després d'un enviament
Via	Informació de passarel·les i servidors <i>proxy</i>
Warning	Informació addicional sobre l'estat d'un missatge

Capçaleres de petició	
capçalera	utilització
Accept	Llista de tipus MIME acceptats pel client
Accept-Charset	Llista de jocs de caràcters acceptats pel client
Accept-Encoding	Tipus de codificació suportats pel client
Accept-Language	Llista de llenguatges acceptats pel client
Authorization	Informació d'autenticació
Expect	Estat esperat
From	Adreça de correu electrònic de l'usuari
Host	Nom i port del host sol·licitat
If-Match	Capçalera utilitzada per a sol·licituds condicionals
If-Modified-Since	Capçalera utilitzada per a sol·licituds condicionals
If-None-Match	Capçalera utilitzada per a sol·licituds condicionals
If-Range	Capçalera utilitzada per a sol·licituds condicionals
If-Unmodified-Since	Capçalera utilitzada per a sol·licituds condicionals
Max-Forwards	Límit de salts per al mètode TRACE
Proxy-Authorization	Informació d'autenticació davant d'un servidor <i>proxy</i>
Range	Rang de <i>bytes</i> sol·licitats
Referer	URL del recurs des del que s'ha realitzat la petició
User-Agent	Informació sobre el navegador del que ha sortit la petició

Capçaleres de resposta	
capçalera	utilització
Allow	Comandes opcionals que també es poden aplicar per a obtenir el recurs
Accept-Ranges	Indica el rang de <i>bytes</i> manegats pel servidor
Age	Estimació de segons que han passat des de l'instant en que es va generar la resposta
Expires	Data en la que el recurs deixa d'estar disponible
Last-Modified	Data local de la última modificació del recurs
Location	Adreça del recurs al qual s'intenta accedir. Permet redirigir sol·licituds
Server	Informació del tipus i versió del servidor HTTP utilitzat
Proxy-Authenticate	Informació d'autenticació davant de servidors <i>proxy</i>
Retry-After	Instant en el que el recurs sol·licitat estarà disponible
Server	Informació sobre el programari utilitzat al servidor
Vary	Informació associada al sistema de cau
WWW-Authenticate	Informació sobre l'accés a un recurs protegit o d'accés restringit

5.1.1.3. RESULTAT DE LA PETICIÓ

Davant de cada petició, el servidor HTTP genera un missatge de resposta on s'insereix una primera línia denominada *línia d'estat* on s'informa sobre el resultat de la operació.

El format d'aquesta primera línia del missatge és la següent:

versió_del_protocol_HTTP codi_estat descripció

És a dir, la versió utilitzada del protocol HTTP, un codi numèric de 3 xifres que indica l'estat de la sol·licitud i un text que conté una breu descripció del codi d'estat retornat. Existeixen 5 categories de missatges d'estat, codificades en funció del primer dígit del codi:

categoria	utilització
1xx	Missatges informatius
2xx	Missatges associats a operacions finalitzades de forma correcta
3xx	Missatges de redirecció
4xx	Missatges associats a errors de client
5xx	Missatges associats a errors de servidor

A la següent taula es poden veure alguns dels missatges més habituals:

missatge	descripció	comentari
200	OK	Operació realitzada correctament
201	Created	Operació realitzada correctament; a més s'ha creat un nou recurs que es troba disponible
202	Accepted	Operació realitzada correctament; a més s'ha creat un nou recurs que no es troba disponible
204	No Content	Operació realitzada correctament; no s'ha generat cap contingut d'interés
301	Moved Permanently	El recurs al qual es pretenia accedir ha estat mogut a una altra ubicació de forma permanent
302	Moved Temporarily	El recurs al qual es pretenia accedir ha estat mogut de forma temporal a una altra ubicació
400	Bad Request	Petició no entesa pel servidor
401	Unauthorized	La petició requereix d'una autenticació
403	Forbidden	Està prohibit l'accés al recurs sol·licitat
404	Not Found	El recurs sol·licitat no es troba al servidor
500	Internal Server Error	S'ha produït un error intern del servidor
501	Not Implemented	El servidor no té implementat algun dels requeriments sol·licitats pel client
503	Service Unavailable	El servidor no es troba disponible

5.1.2. FUNCIONS PHP RELACIONADES

PHP proporciona les següents funcions per a obtenir la informació continguda a les capçaleres HTTP:

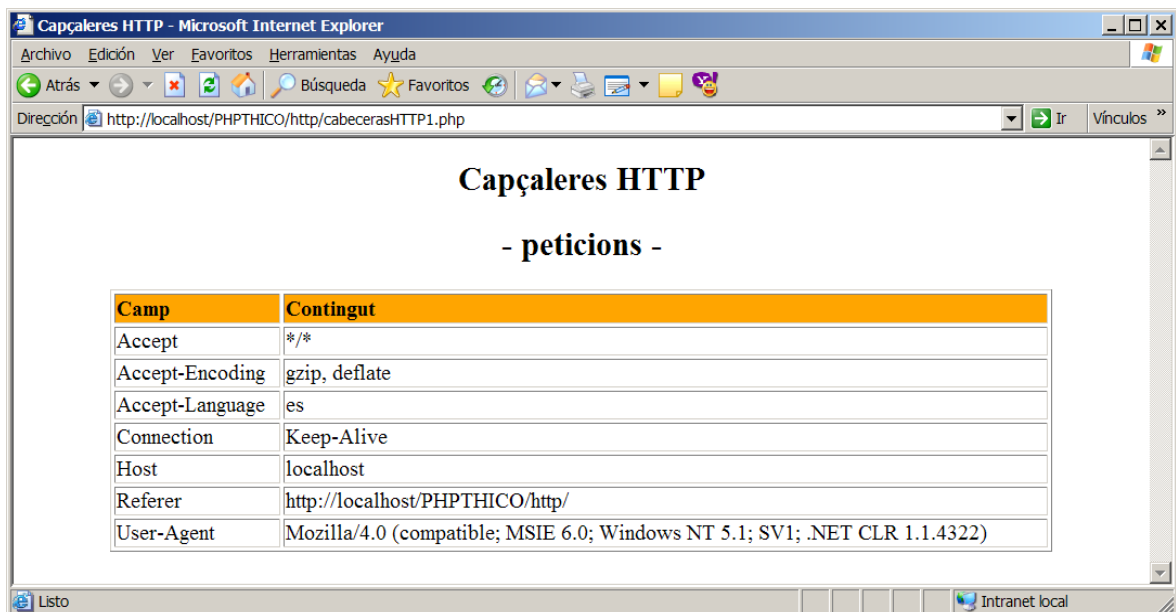
- `apache_request_headers()` per als missatges de petició
- `apache_response_headers()` per als missatges de resposta.

Les dues funcions retornen un array associatiu que conté totes les capçaleres HTTP de la petició/resposta de la operació actual.

NOTA: Aquestes funcions només estan disponibles quan s'està executant PHP com a mòdul d'Apache.

Al següent exemple podem veure com recuperar les capçaleres de petició:

```
<HTML>
<HEAD>
  <TITLE>Capçaleres HTTP</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <H2>Capçaleres HTTP</H2>
    <H2>- peticions -</H2>
    <?php
      //$cabecera=getallheaders();
      $cabecera=apache_request_headers();
      echo "<TABLE BORDER='1' WIDTH='85%'><TR BGCOLOR='orange'>";
      echo "<TD><B>Camp</B></TD><TD><B>Contingut</B></TD></TR>";
      foreach($cabecera as $campo => $contenido)
        echo "<TR><TD>$campo</TD><TD>$contenido</TD></TR>";
      echo "</TABLE>";
    ?>
  </CENTER>
</BODY>
</HTML>
```



PHP també proporciona funcions per a la gestió de les capçaleres:

- ◆ `header()`: per a enviar capçaleres HTTP

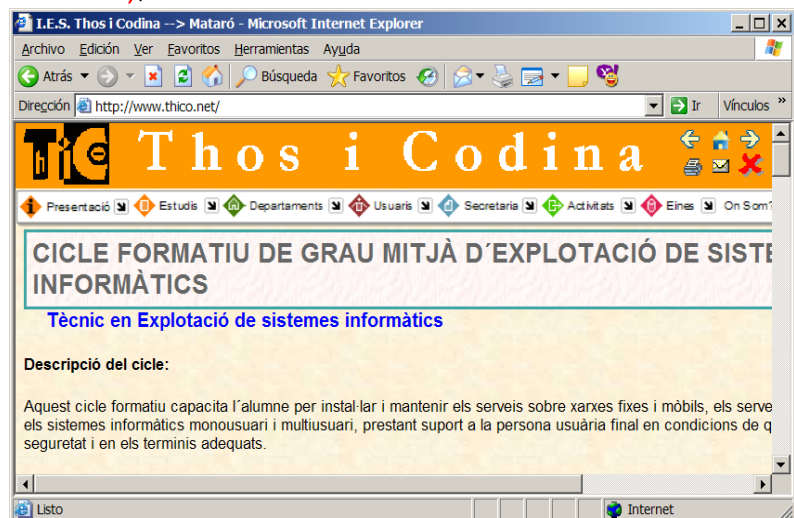
`header(cadena [, reemplaçar [, http_response_code]])`

- ◆ `headers_sent()`: per a confirmar que les capçaleres han estat enviades correctament.

El primer paràmetre opcional de la funció `header()`, de caràcter booleà, permet indicar si la capçalera s'ha de reemplaçar per una capçalera similar prèviament fixada o, pel contrari, ha d'afegir una nova capçalera del mateix tipus (el valor per defecte és `true`). El segon paràmetre opcional, un valor enter, força el codi de resposta que s'ha d'enviar en processar la sol·licitud.

Un dels usos més habituals de la funció `header()` consisteix en enviar la capçalera `Location` al navegador. D'aquesta forma es poden redirigir les peticions enviades al nostre servidor.

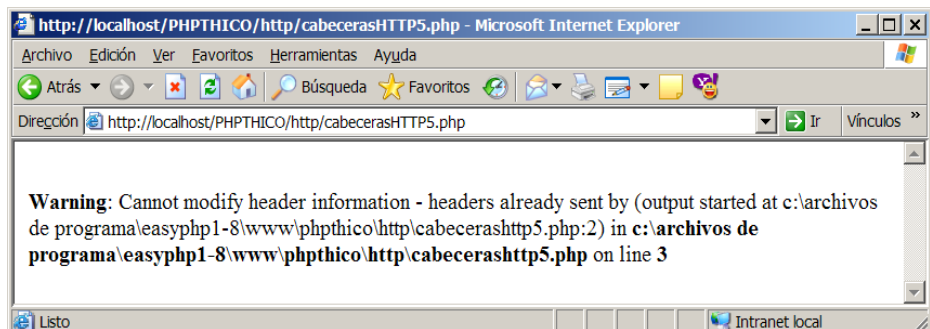
```
<?php
header("Location: http://www.thico.net");
?>
```



La funció `header()` ha d'aparèixer abans de que es generi qualsevol tipus de contingut en el document, ja que definirà un component de les capçaleres HTTP del missatge. Si a l'exemple anterior li afegim un espai en blanc davant de la primera etiqueta de php:

```
1
2 <?php
3 header("Location: http://www.thico.net");
4 ?>
```

El resultat serà:



5.1.3. VARIABLES PHP RELACIONADES

Les variables PHP directament relacionades amb la informació manegada pel protocol HTTP són:

- | | | | |
|---|------------------------|---|-------------------------|
| x | <code>\$_SERVER</code> | x | <code>\$_POST</code> |
| x | <code>\$_GET</code> | x | <code>\$_REQUEST</code> |

Les 3 últimes les veurem amb més deteniment quan parlem de formularis.

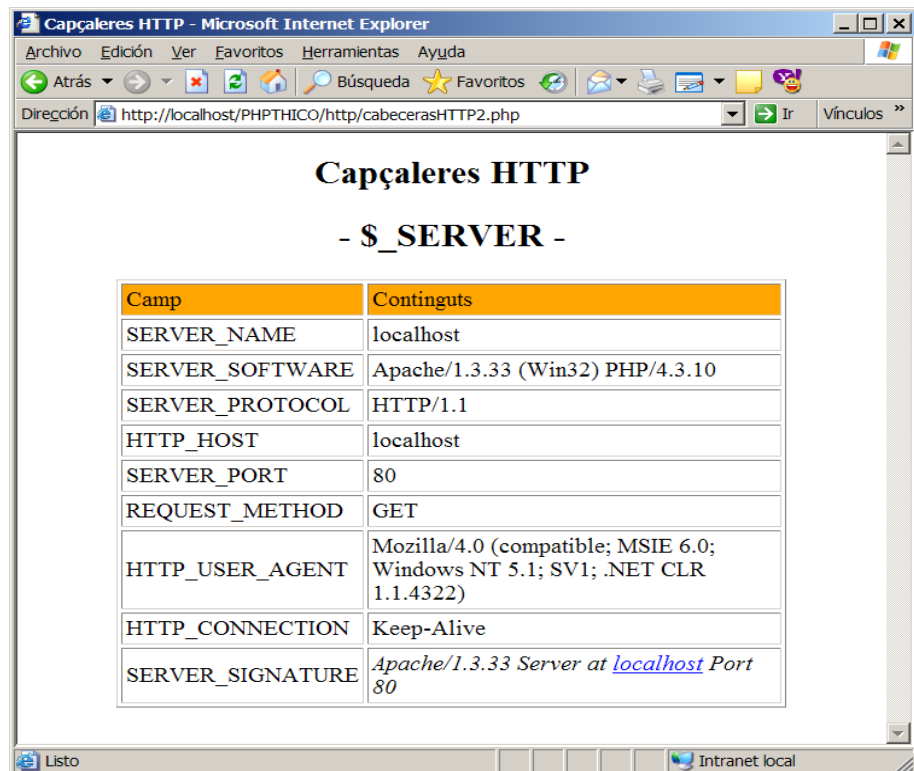
La variable global **`$_SERVER`** és un array associatiu que conté, entre altres coses, tota la informació de les capçaleres tant de petició com de resposta. El contingut d'aquesta variable es pot veure a:

camp	descripció
PHP_SELF	Nom de l'script PHP que s'està executant
argv	Array amb els paràmetres passats com a arguments a l'script
argc	Nombre de paràmetres passats com a arguments a l'script
GATEWAY_INTERFACE	Especificació del CGI que utilitza el servidor
SERVER_NAME	Nom de l'equip servidor
SERVER_SOFTWARE	Identificació del programari que està utilitzant el servidor
SERVER_PROTOCOL	Nom i versió del protocol utilitzat pel servidor
REQUEST_METHOD	Mètode utilitzat per la sol·licitud per accedir als recursos
QUERY_STRING	Cadena de petició
DOCUMENT_ROOT	Directorí arrel del document en el que es troba l'script
HTTP_ACCEPT	Continguts de la capçalera <i>accept</i>
HTTP_ACCEPT_CHARSET	Continguts de la capçalera <i>accept-charset</i>
HTTP_ENCODING	Continguts de la capçalera <i>accept-encoding</i>
HTTP_ACCEPT_LANGUAGE	Continguts de la capçalera <i>accept-language</i>
HTTP_CONNECTION	Continguts de la capçalera <i>connection</i>
HTTP_HOST	Continguts de la capçalera <i>host</i>
HTTP_REFERER	Adreça de la pàgina de la qual es procedeix
HTTP_USER_AGENT	Continguts de la capçalera <i>user-agent</i>
REMOTE_ADDR	Adreça IP de l'usuari
REMOTE_PORT	Port utilitzat per la màquina de l'usuari
SCRIPT_FILENAME	Camí d'accés a l'script que s'està executant
SERVER_ADMIN	Valor de la directiva <code>SERVER_ADMIN</code> de l'Apache
SERVER_PORT	Port utilitzat per l'equip servidor
SERVER_SIGNATURE	Cadena que identifica al servidor
PATH_TRANSLATED	Camí d'accés a l'script tenint en compte el sistema de fitxers
SCRIPT_NAME	Camí d'accés a l'script actual
REQUEST_URI	URI utilitzada per accedir al recurs
PHP_AUTH_USER	Variable utilitzada per a realitzar l'autenticació d'usuaris amb HTTP. Emmagatzema el nom de l'usuari. Només funciona quan PHP s'executa com a mòdul d'Apache.
PHP_AUTH_PWD	Variable utilitzada per a realitzar l'autenticació d'usuaris amb HTTP.

camp	descripció
	Emmagatzema la clau de l'usuari. Només funciona quan PHP s'executa com a mòdul d'Apache.
PHP_AUTH_TYPE	Variable utilitzada per a realitzar l'autenticació d'usuaris amb HTTP. Emmagatzema el tipus d'autenticació. Només funciona quan PHP s'executa com a mòdul d'Apache.

En el següent exemple es mostra la utilització de la variable \$_SERVER per a recuperar informació sobre les capçaleres HTTP:

```
<?php
echo "SERVER_NAME: $_SERVER[SERVER_NAME]";
echo "SERVER_SOFTWARE: $_SERVER[SERVER_SOFTWARE]";
echo "SERVER_PROTOCOL: $_SERVER[SERVER_PROTOCOL]";
echo "HTTP_HOST: $_SERVER[HTTP_HOST]";
echo "SERVER_PORT: $_SERVER[SERVER_PORT]";
echo "REQUEST_METHOD: $_SERVER[REQUEST_METHOD]";
echo "HTTP_USER_AGENT: $_SERVER[HTTP_USER_AGENT]";
echo "HTTP_CONNECTION: $_SERVER[HTTP_CONNECTION]";
echo "SERVER_SIGNATURE: $_SERVER[SERVER_SIGNATURE]";
?>
```



Un ús habitual de les capçaleres HTTP utilitzant els camps **PHP_AUTH_USER** i **PHP_AUTH_PW** de la variable `$_SERVER` permet realitzar l'autenticació bàsica d'usuaris per a l'accés a recursos restringits.

```
<?php
if (!isset($_SERVER[PHP_AUTH_USER])) {
    header("WWW-Authenticate: Basic realm="WebPHP");
    header('HTTP/1.0 401 Unauthorized');
    echo '<CENTER><H2>Accés restringit...</H2>';
    echo '<HR>És necessari tenir autorització per a accedir a aquest recurs.<BR>';
    echo '<BR>Posi's en contacte amb l' ';
    echo '<A HREF="mailto:'. $_SERVER[SERVER_ADMIN].'">administrador</A><BR><HR>';
    echo '</CENTER>';
    exit;
} elseif ((($_SERVER[PHP_AUTH_USER]!='libroPHP')||
                                                    ($_SERVER[PHP_AUTH_PW]!='acceso')) {
    header("WWW-Authenticate: Basic realm="WebPHP");
    header('HTTP/1.0 401 Unauthorized');
    echo '<CENTER><H2>Accés restringit...</H2>';
    echo '<HR>És necessari tenir autorització per a accedir a aquest recurs.<BR>';
    echo '<BR>Posi's en contacte amb l' ';
    echo '<A HREF="mailto:'. $_SERVER[SERVER_ADMIN].'">administrador</A><BR><HR>';
    echo '</CENTER>';
    exit;
} else {
    // Accés concedit...
    echo '<CENTER><H2>Accés concedit...</H2></CENTER>';
}
?>
```

