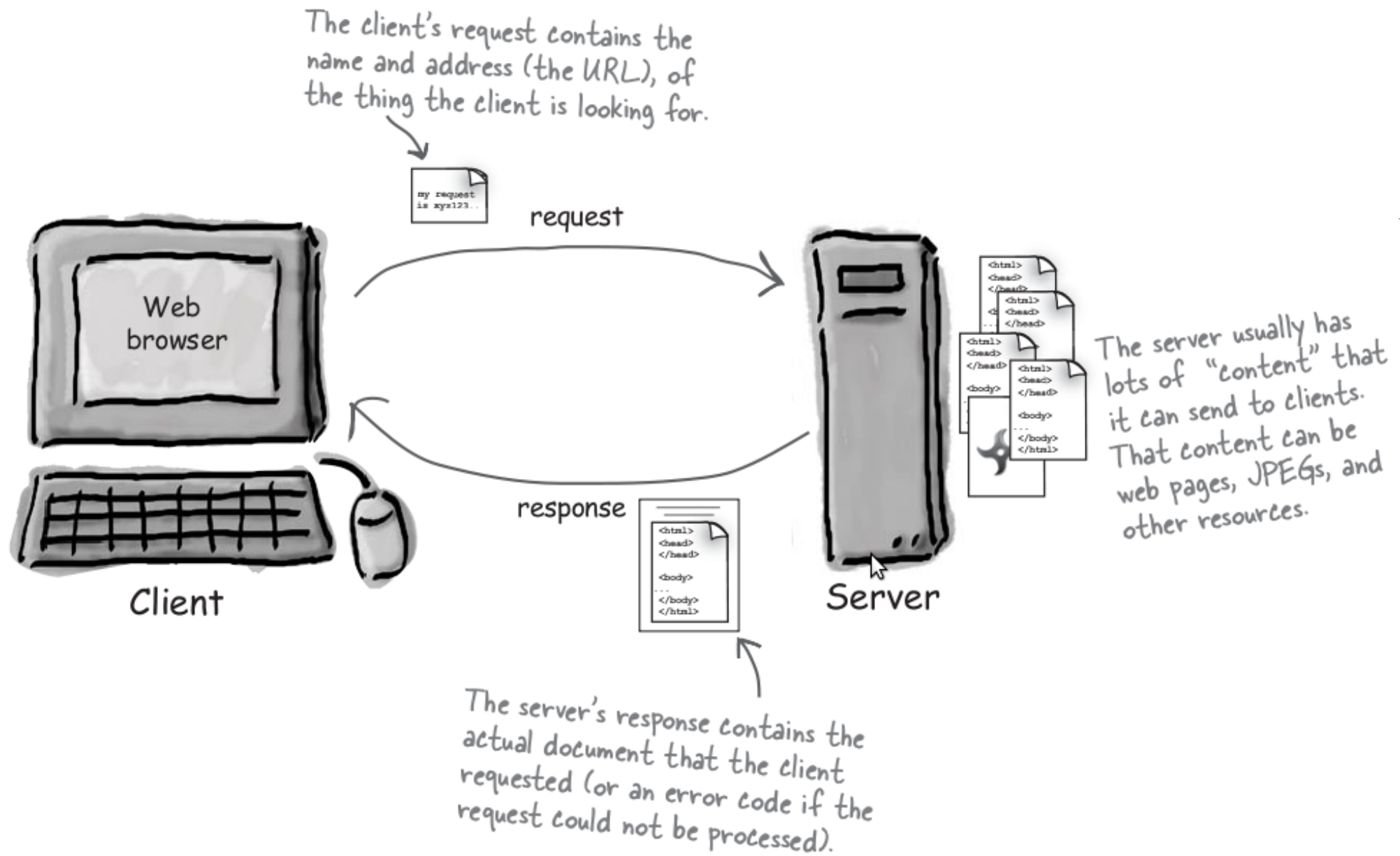
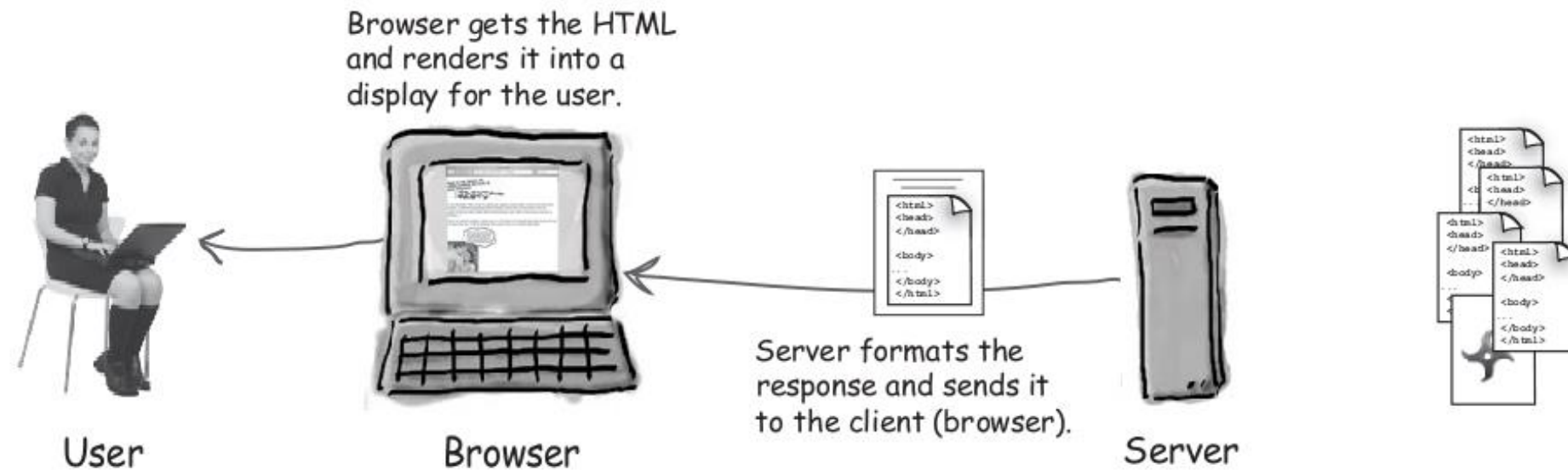
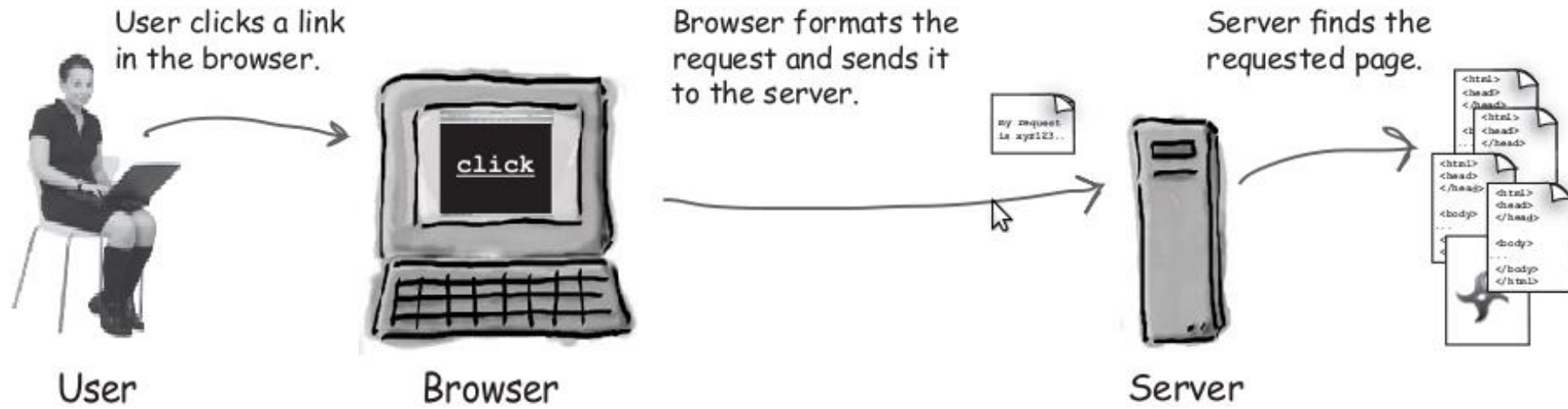


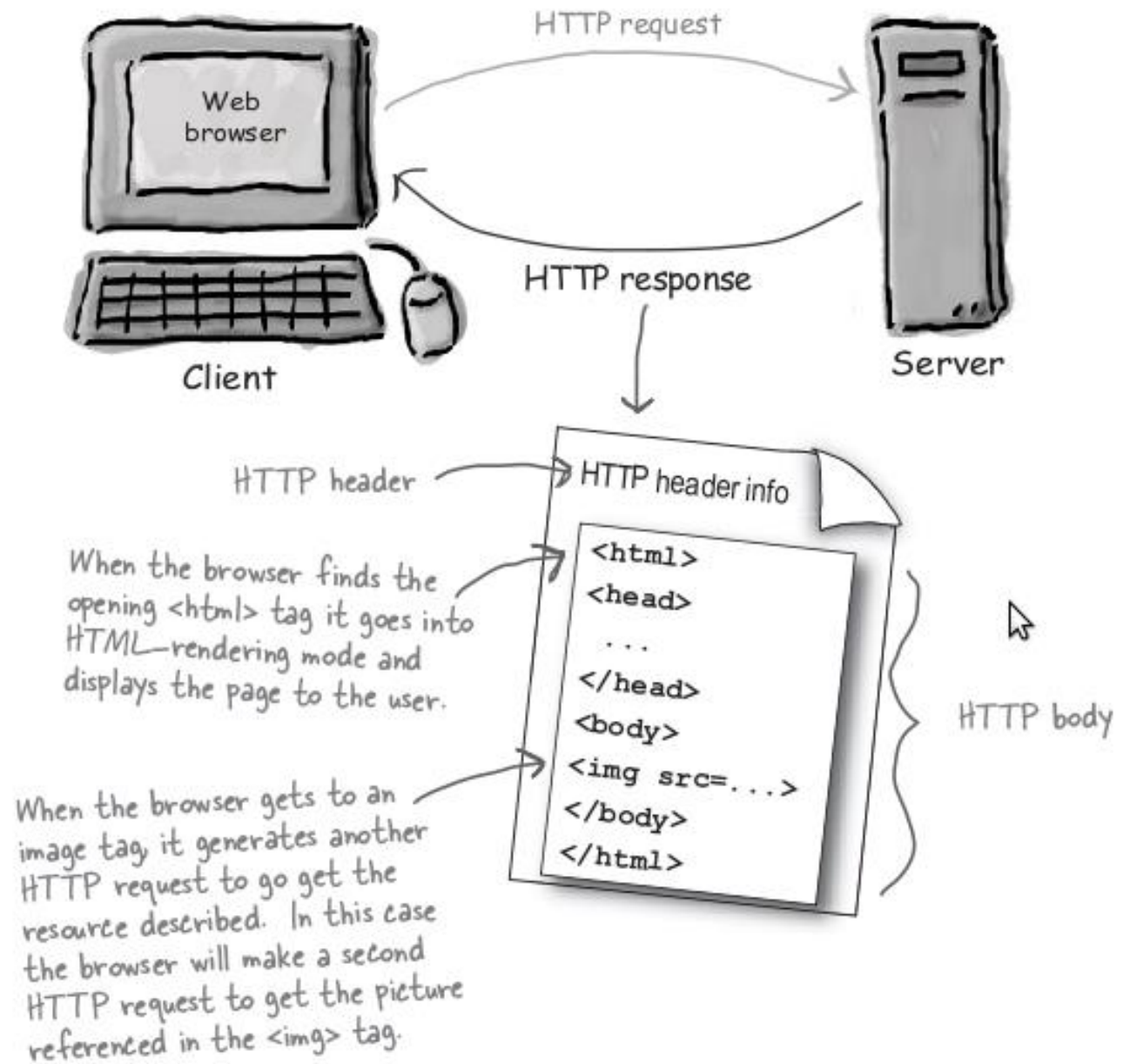
Protocolo HTTP

VÍCTOR CUSTODIO

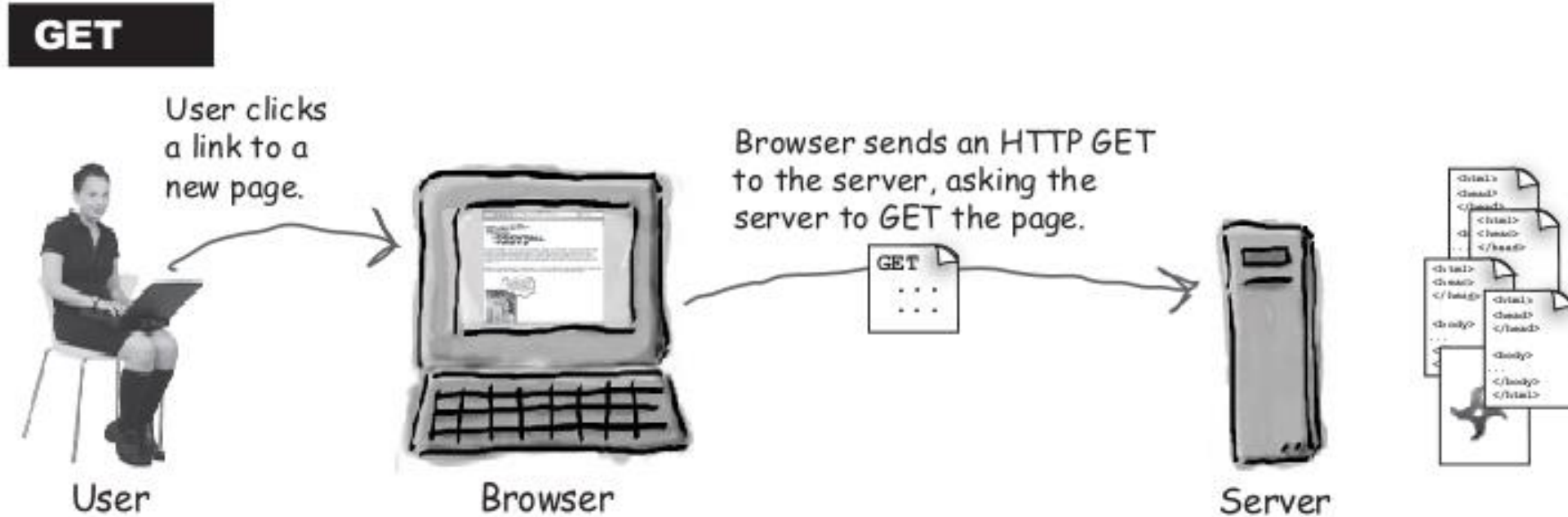




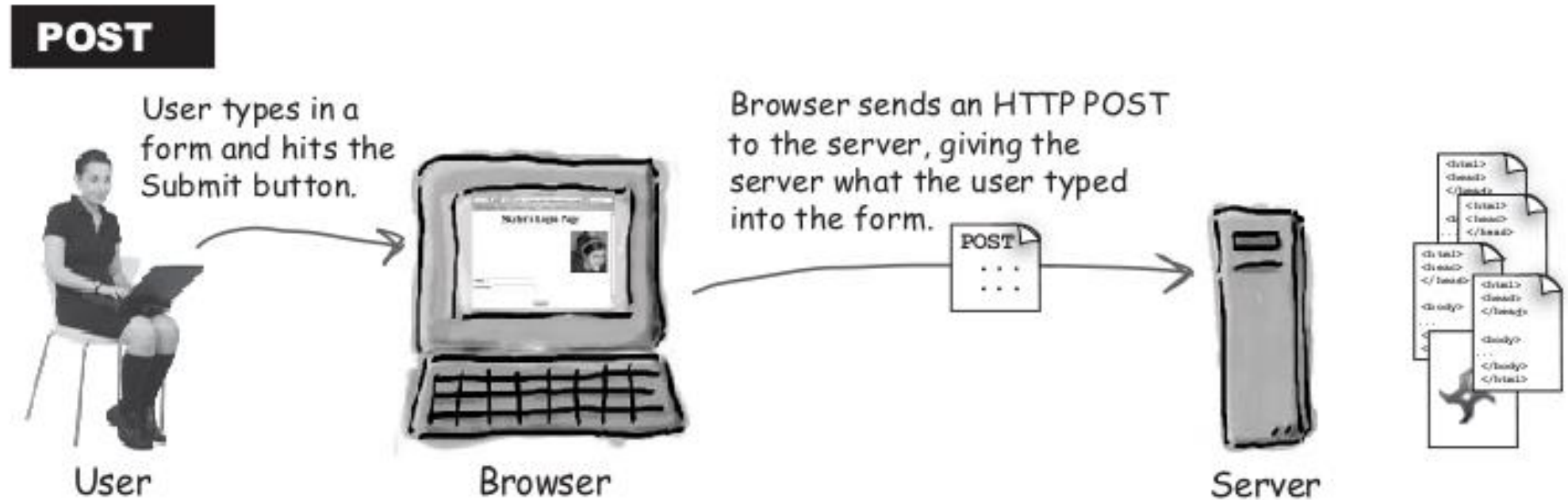
El Código HTML es parte del mensaje de respuesta HTTP



Metodo GET



Método POST



Terminología

Una página Web es un conjunto de objetos.

- Los objetos pueden ser archivos HTML, imágenes JPEG, applets de Java, archivos de audio, etc.
- Una página Web consiste en un archivo base HTML el cual incluye referencias a varios objetos
- Cada objeto es alcanzable por un URL.

Ejemplo:

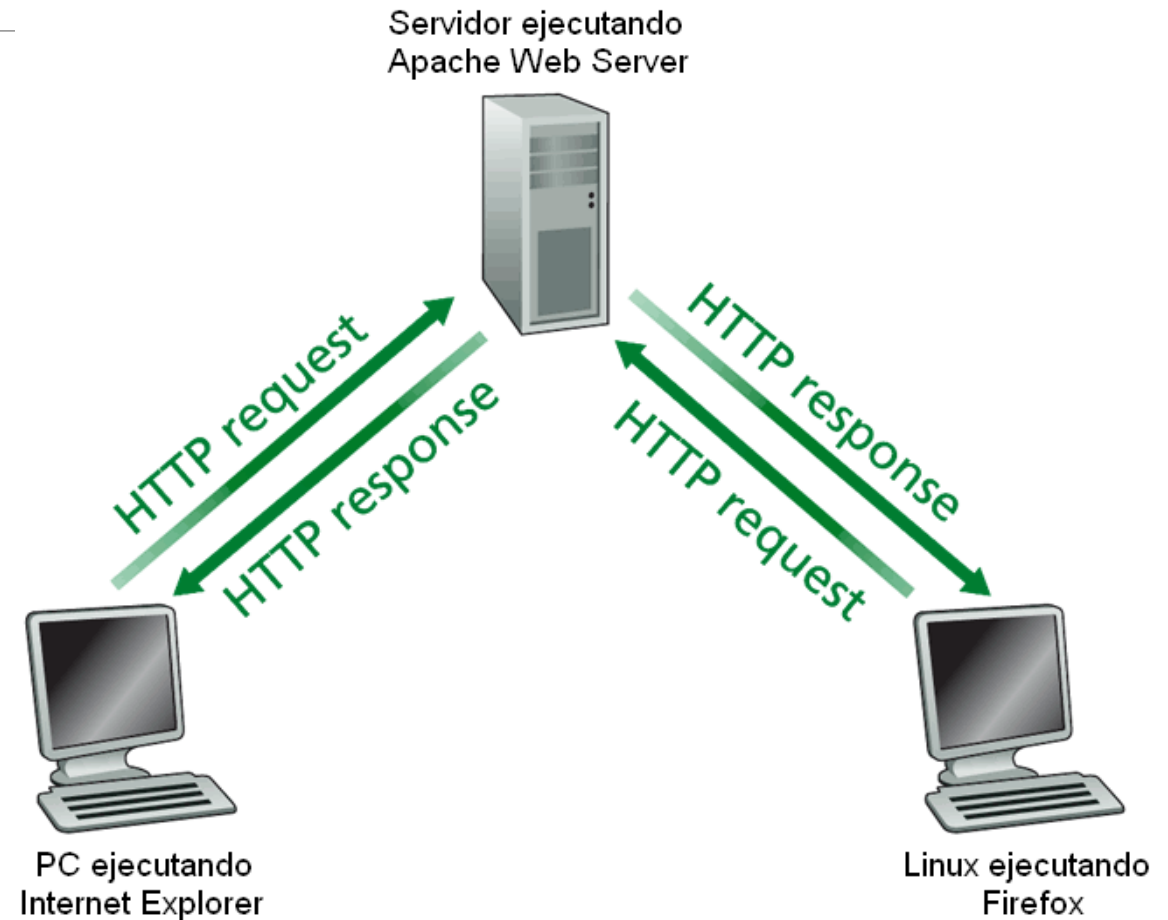
`grid.uniquindio.edu.co/proyecto437/informe.pdf`

Host name Path name

Generalidades de HTTP

HTTP: hypertext transfer protocol.

- Protocolo de la capa de aplicación.
- Modelo cliente/servidor.
 - Cliente: navegador que solicita, recibe y muestra los objetos web.
 - Servidor: El servidor Web envía objetos en respuesta a las solicitudes.
- HTTP 1.0: RFC 1945.
- HTTP 1.1: RFC 2068, 2616



Generalidades de HTTP

Usa TCP.

- El cliente inicia la conexión (crea el socket) al servidor, en el puerto 80.
- El servidor acepta la conexión TCP del cliente.
- Intercambio de mensajes de la capa de aplicación entre el navegador (cliente HTTP) y el servidor Web (servidor HTTP).

Generalidades de HTTP

- La conexión TCP se cierra al terminar la comunicación.
- El servidor no mantiene información del estado acerca de las solicitudes anteriores del cliente.

Mensajes HTTP

Dos tipos de mensajes HTTP

- HTTP Request
- HTTP Response

HTTP Request Message

Mensaje HTTP Request:

- ASCII (formarlo legible)

Línea de request
(comandos GET, POST,
HEAD)

Líneas de encabezado

Carriage return,
line feed
Indica fin de mensaje

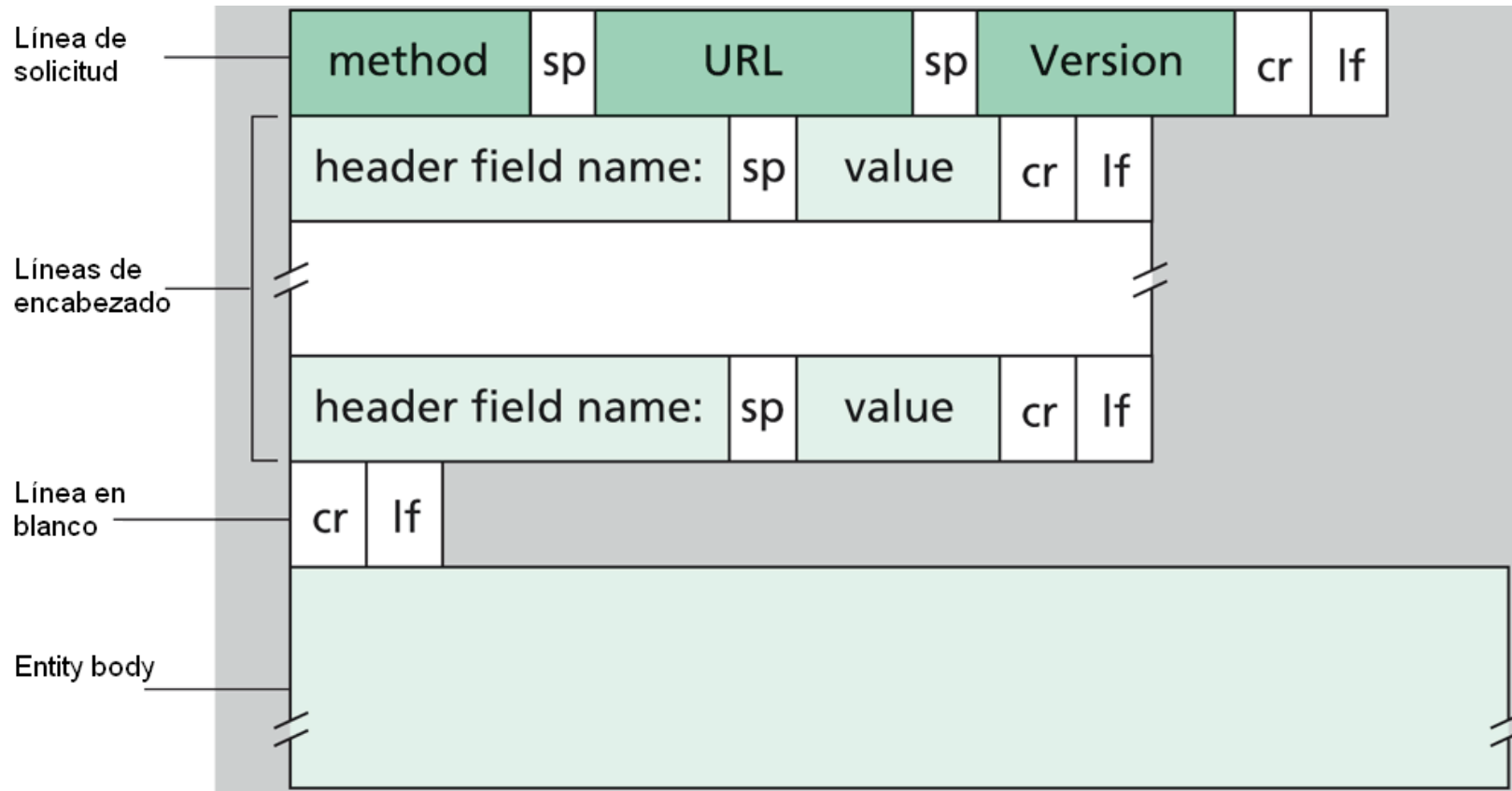
```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

(carriage return, line feed extra)

HTTP Request Message

```
GET /proyecto437/index.html HTTP/1.1\r\n
Host: grid.uniquindio.edu.co\r\n
Connection: Keep-Alive\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0)\r\n
Accept-Language: es-co\r\n
\r\n
```

HTTP Request Message



HTTP Response

Línea de estatus
(código de estatus
del protocolo
Frase de estatus)

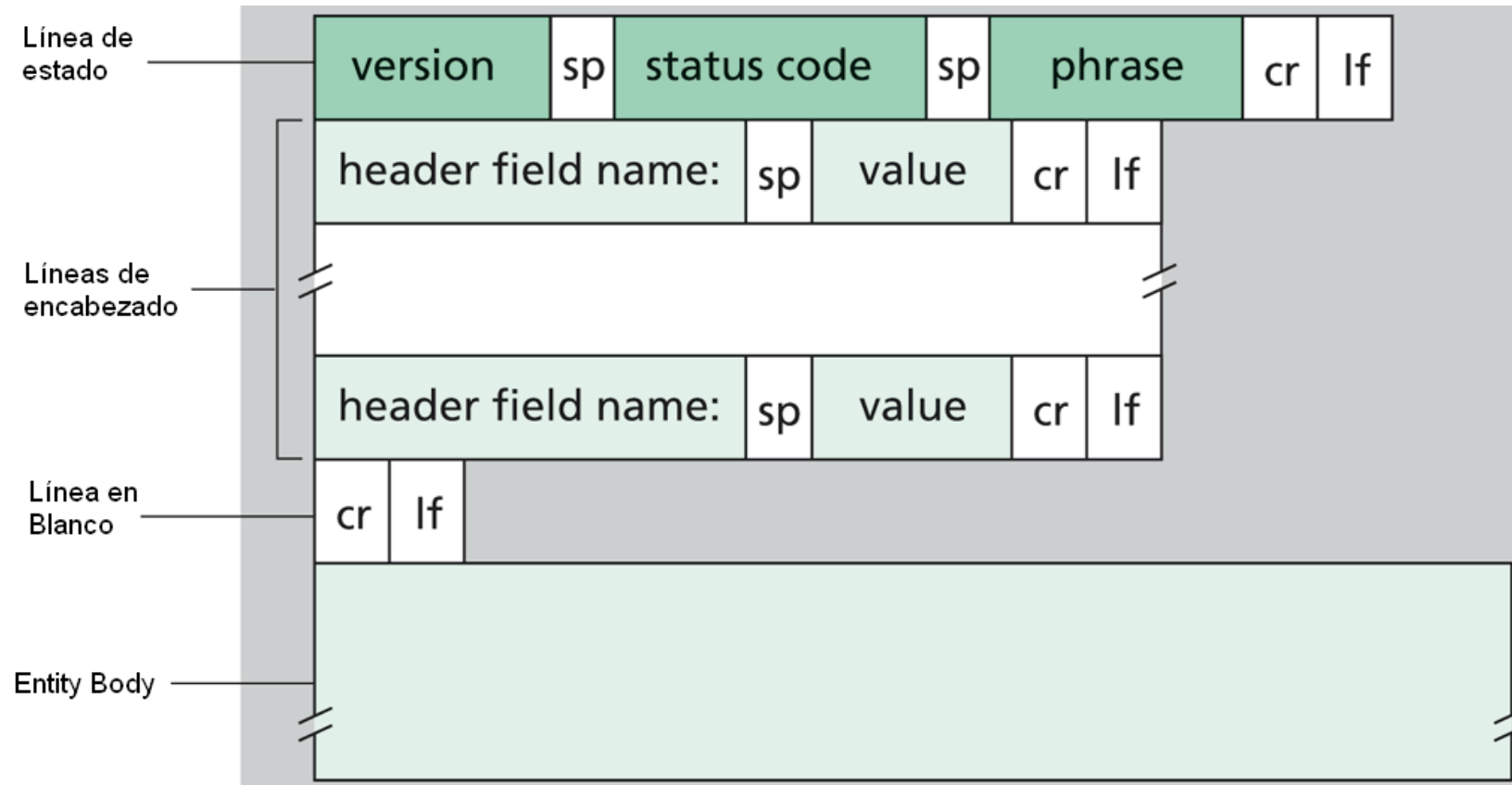
Líneas de
encabezado

data, e.g.,
archivo
HTML solicitado

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html

data data data data data ...
```

HTTP Response



HTTP Response

```
HTTP/1.1 200 OK\r\n
Date: Wed, 07 Oct 2009 20:09:43 GMT\r\n
Server: Apache/1.3.0 (Unix)\r\n
Last-Modified: Tue, 1 Sep 2009 14:29:17 GMT\r\n
Content-Length: 4565\r\n
Content-Type: text/html\r\n
data data data data data ...
```

Códigos de Respuesta

- **200 OK**

- Solicitud exitosa, el objeto solicitado va después en este mensaje

- **301 Moved Permanently**

- El objeto solicitado ha sido movido, la nueva ubicación será especificada después en este mensaje (Location:)

- **400 Bad Request**

- El mensaje de solicitud no fue entendido por el servidor

- **404 Not Found**

- El objeto solicitado no fue encontrado en este servidor

- **505 HTTP Version Not Supported**

- La versión del protocolo HTTP no es soportada por este servidor

Tipos de Métodos

HTTP/1.0

GET

POST

HEAD

- Pide al servidor dejar el objeto requerido fuera de la respuesta.(solo la cabecera)

HTTP/1.1

GET, POST, HEAD

PUT

- Sube archivos en cuerpo del requerimiento en la localización indicada por el campo URL

DELETE

- Borra archivo especificado en el campo URL

Ejemplo protocolo Http con sockets

```
public class ClienteHttp {
public ClienteHttp(){

String peticion;
DataOutputStream salida;
DataInputStream entrada;
String cad="";
byte datosBytes[]=new byte[256];
int leido=0;
try{
    Socket conexion=new Socket ("localhost", 80);
    System.out.println("Conectando");
    /*Enviamos lo que envía el navegador Internet Explorer al pedir una
    página*/
    peticion="GET /index.html.es HTTP/1.1\n";
    peticion+="Accept: /**\n";
    peticion+="Accept-Language: es\n";
    peticion+="Accept-Encoding: gzip, deflate\n";
    peticion+="User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows
NT)\n";
```

Ejemplo protocolo Http con sockets

```
peticion+="Host: 127.0.0.1\n";
peticion+="Connection: Keep-Alive\n\n";
salida=new DataOutputStream(conexion.getOutputStream());
salida.write(peticion.getBytes());
//Vemos lo que nos envia el Servidor
entrada=new DataInputStream(conexion.getInputStream());
while ((leido=entrada.read(datosBytes,0,256))!=1)

    if (leido>0)
        System.out.println (new String(datosBytes,0,(leido-1)));

} catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

}
public static void main (String args[]){
ClienteHttp s = new ClienteHttp();
}
}
```

Estado usuario-servidor: cookies

La mayoría de sitios utilizan Cookies

Cuatro componentes:

- 1) Línea encabezado cookie en el mensaje respuesta HTTP
- 2) Línea de encabezado cookie en el request HTTP
- 3) Archivo cookie es almacenado en la máquina del usuario y administrada por su navegador.
- 4) Base de datos en sitio Web

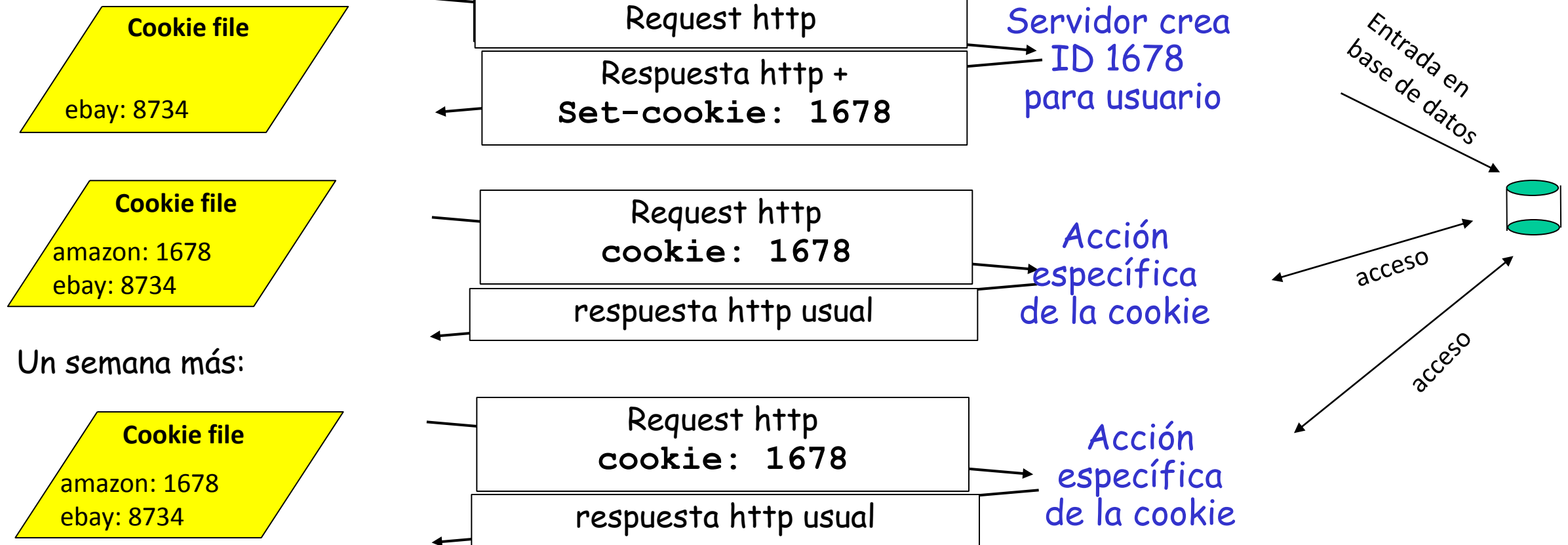
Ejemplo:

- Susan accede Internet siempre desde el mismo PC
- Ella visita un sitio e-commerce específico por primera vez.
- Cuando el request HTTP inicial llega al sitio, éste crea un ID único y crea una entrada en la base de datos para ese ID.

Cookies: conservando el “estado” (cont.)

cliente

servidor



Cookies (cont.)

Al margen

Qué pueden transportar las cookies:

- autorización
- shopping carts
- sugerencias
- Estado de la sesión del usuario (Web e-mail)

Cookies y privacidad:

- ☐ Las cookies permiten que un sitio aprenda mucho sobre ti.
- ☐ Podríamos proveer nombre y correo al sitio si tenemos cookies que lo almacenen.

Ejercicio

Crear un servidor web en Java que devuelva por http la siguiente página web si se le realiza un request GET solicitando index.html

```
<html>;  
<body>;  
    <h1>Hola que tal!</h1>  
</body>  
</html>;
```

Y que devuelva una imagen.jpg si se le solicita /imagen.jpg

Si la solicitud no es ninguna de estas, debe devolver un error indicando que no existe el recurso

El encabezado Http debe llevar al menos los siguientes valores:

Date , content-type, content-length.

Anexo : códigos de respuesta

1xx Mensajes

100

111 Conexión rechazada

2xx Operación exitosa

2xx Mensajes

200 OK

201-203 Información no oficial

204 Sin Contenido

205 Contenido para recargar

206 Contenido parcial 3xx Redirección

Anexo : códigos de respuesta

3xx Redirección

301 Mudado permanentemente

302 Encontrado

303 Vea otros

304 No modificado

305 Utilice un proxy

307 Redirección temporal

Anexo : códigos de respuesta

4xx Error por parte del cliente

400 Solicitud incorrecta

401 No autorizado

402 Pago requerido

403 Prohibido

404 No encontrado

409 Conflicto

410 Ya no disponible

412 Falló precondition

Anexo : códigos de respuesta

5xx Error del servidor

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

Anexo : encabezados http

Encabezado	Significado
Cache-Control	Permite especificar distintas directivas para controlar la caché, tanto del cliente como de servidores proxy.
Connection	Especifica opciones de la conexión de red.
Date	Envía una fecha en la representación estándar definida en el protocolo.
Pragma	Transporta información no HTTP a un receptor que sea capaz de entenderla.
Trailer	Indica que ciertas cabeceras HTTP pueden encontrarse en el final de un mensaje con múltiples partes (multipart).
Upgrade	Información sobre protocolos adicionales soportados por el cliente.
Via	Añadidos al mensaje de proxys o gateways para indicar que pasó por ellos.
Warning	Información adicional sobre un estado o transformación de un documento que podría no estar reflejado en el cuerpo del mismo. Por ejemplo, transformaciones que se hacen en servidores caché.

Anexo : encabezados http Request

Encabezado	Significado
Accept	Listado de tipos MIME que el cliente soporta. Hay otros encabezados relacionados como Accept-Charset , Accept-Encoding , Accept-Language .
Authorization	Indica las credenciales de acceso a un recurso que presenta el usuario.
Expect	Indica que comportamiento del servidor necesita el cliente.
From	Dirección de correo que controla el cliente (navegador).
Host	Nombre o IP del host desde donde se conecta el cliente.
If-Match	Un cliente que tiene recursos en cache puede verificar si están actualizados incluyendo este encabezado. Hay otros encabezados que también tienen que ver con la caché. If-Modified-Since , If-None-Match , If-Range , If-Unmodified-Since .
Max-Forwards	Cuantas veces la petición del cliente puede ser reenviada por proxys.
Proxy-Authoritation	Indica las credenciales de acceso a un proxy que presenta el usuario.
Range	Indica que porción de recurso (rango de bytes) recuperar.
Referer	Es el URI del recurso desde donde la petición se ha realizado (generalmente por provenir de un enlace HTML).
TE	Que codificaciones de transferencia está dispuesto a recibir el cliente.
User-Agent	Información sobre el agente de usuario (generalmente navegador) que origina la petición.