

Teleinformática y Redes 2016

Protocolo HTTP 1.1
RFC 2616

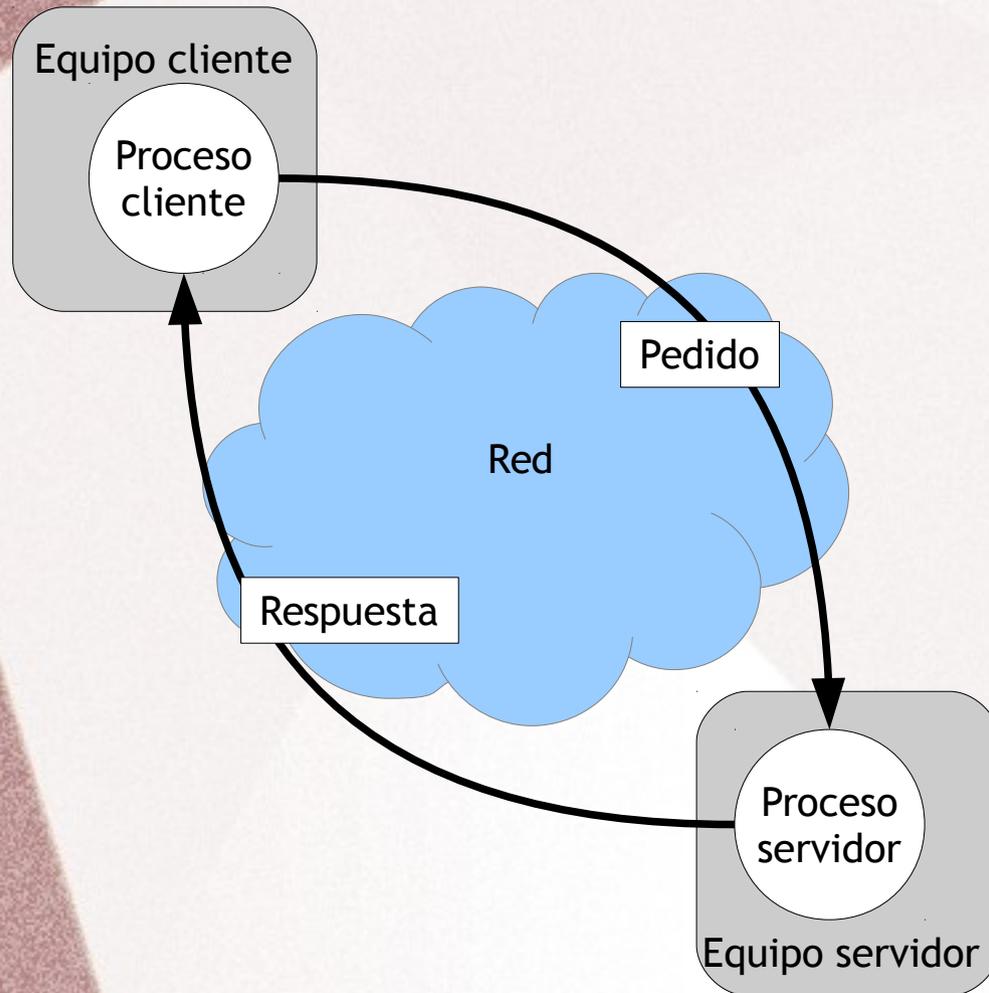
¿Por qué ver este protocolo?

- HTTP comprende alrededor del 53% del tráfico en Internet de América Latina.
- Esta cifra crece a medida que más aplicaciones trasladan su forma de operación a la web (ej: *webchat*).

Application	Share
Youtube	30.11 %
HTTP	12.18 %
BitTorrent	11.65 %
SSL - Other	10.31 %
Netflix	5.91 %
Facebook	5.66 %
MPEG - Other	2.14 %
RTMP	1.82 %
Flash Video	1.70 %
Other (email, ftp, etc)	18.52 %

Latin America, Fixed Access, Peak Period, Top Apps by Bytes
Global Internet Phenomena Report: 2015, Sandvine Inc. ULC
<https://www.sandvine.com/downloads/general/global-internet-phenomena/2015/global-internet-phenomena-report-latin-america-and-north-america.pdf>

Arquitectura cliente/servidor



- En la arquitectura **cliente/servidor**, el cliente es un componente que realiza peticiones de servicio a otro componente, el servidor, que satisface dichas peticiones. Ambos componentes están conectados mediante algún tipo de red.

HyperText Transfer Protocol

HTTP 1.0 - IETF '96 - RFC 1945

- Se crea una pagina web mediante HTML, alojada como un archivo en el servidor web, en un directorio publico. El usuario escribe una dirección web en su cliente web o *User Agent (UA)*, el UA establece una conexion con el servidor web y solicita el recurso.
- El servidor web busca el archivo y, si existe, lo remite al cliente. Por cada objeto requerido se realiza una nueva petición independiente de la anterior.

Características de HTTP

- **Capa 7** Opera en capa de Aplicación sobre transporte TCP
- **Petición/Respuesta.** Basa su operación en transacciones.
- **Sin estado.** Cada transacción HTTP es independiente de la anterior y de la siguiente.
- **Bidireccional.** Permite al servidor web enviar datos al cliente y viceversa.
- **Negocia capacidades.** Permite al UA negociar características tales como tipo de contenido aceptados, idioma, etc
- **Intermediarios y caché.** Permite que estaciones actúen como intermediarios y que se almacenen y recuperen copias de objetos para mejorar los tiempos de respuesta.

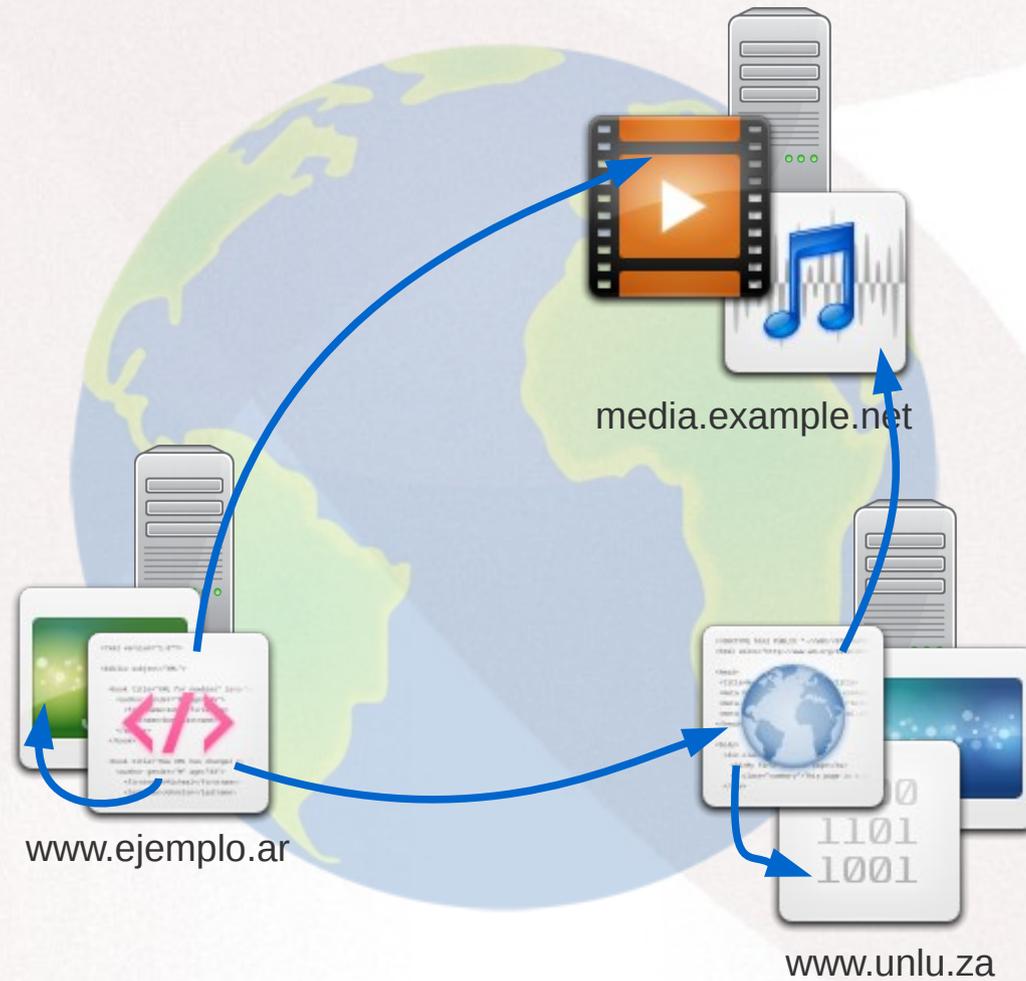
HyperText Markup Language

HTML - CERN '91

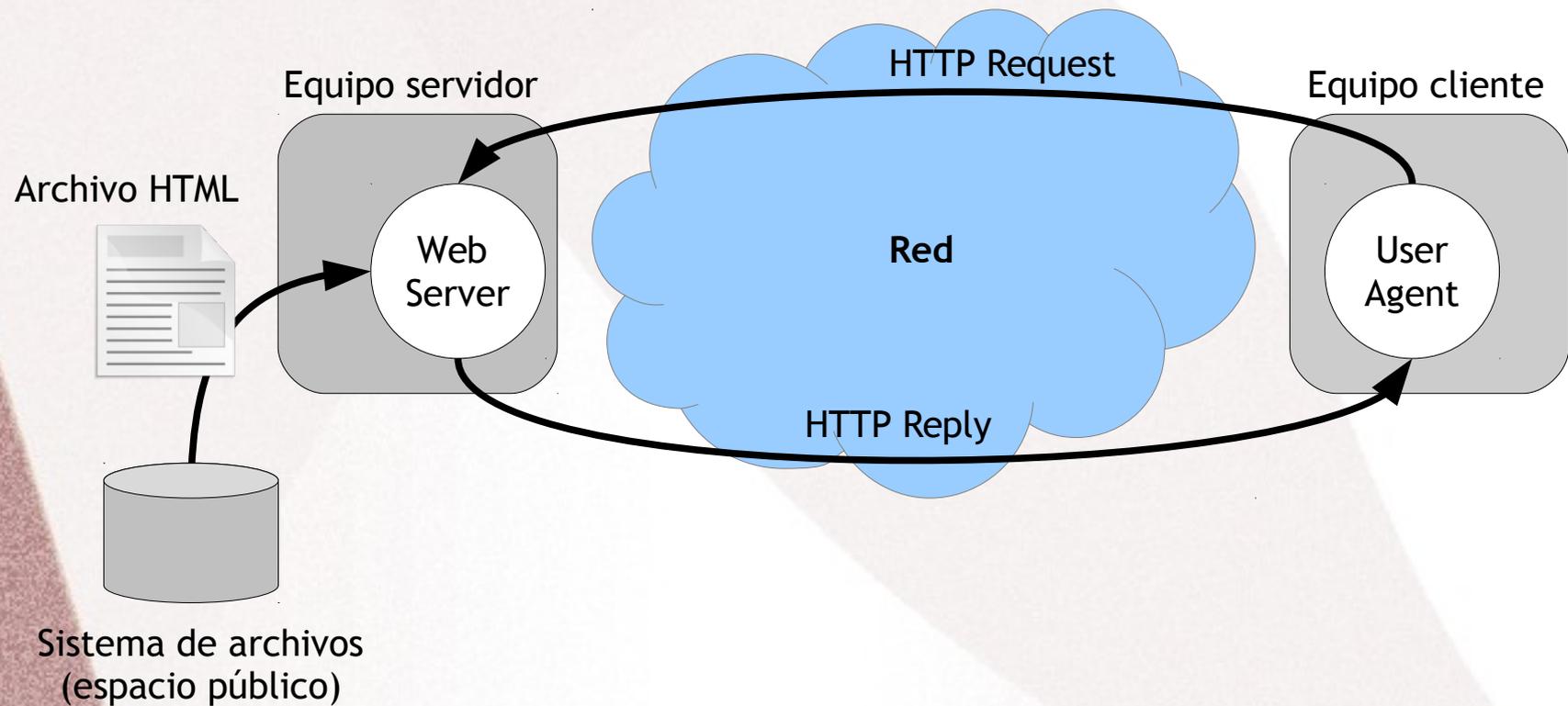
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01//EN" "http://www.w3.org/
TR/html4/strict.dtd">
<html>
<head>
  <title>Planet Express</title>
</head>
<body bgcolor="#CCCCCC">
  <center>
    
    <h1>Planet Express</h1>
    <p>Bienvenido a Planet Express:
repartos a todo el Universo.</p>
    <p>
      <a href="envios.html">
Envios</a> /
      <a href="staff.html">
Staff</a>
    </p>
  </center>
</body>
</html>
```

- Es un lenguaje de etiquetas que permite crear documentos estructurados especificando la semántica del texto, tal como párrafos, encabezados, listas, enlaces y tablas. Permite embeber imágenes, formularios y otros objetos. Los documentos se escriben en texto plano con etiquetas o *tags* entre signos `<` y `>`

La Web como sistema distribuido



Páginas estáticas

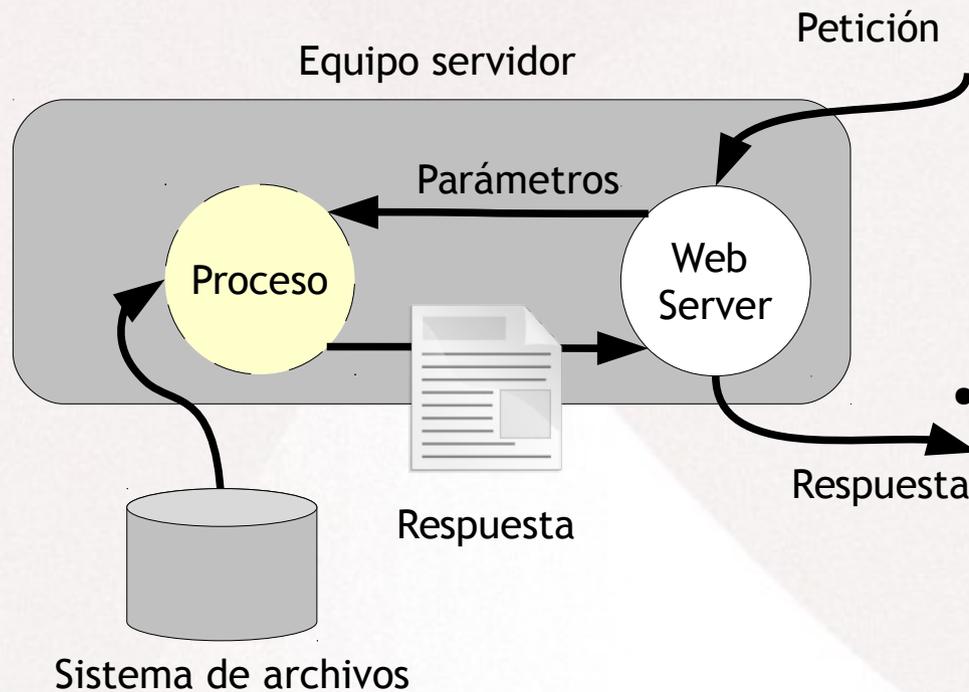


Páginas estáticas

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01//EN" "http://www.w3.org/
TR/html4/strict.dtd">
<html>
<head>
  <title>Planet Express</title>
</head>
<body bgcolor="#CCCCCC">
  <center>
    
    <h1>Planet Express</h1>
    <p>Bienvenido a Planet Exp
repartos a todo el Universo.</
<p>
    <a href="envios.html">
Envios</a> /
    <a href="staff.html">
Staff</a>
    </p>
  </center>
</body>
</html>
```



Procesamiento del lado del servidor



- Páginas HTML que se generan mediante un proceso auxiliar cada vez que un cliente las solicita. De ahí el nombre **páginas dinámicas**.
- La comunidad debió acordar una interfaz de comunicación entre el proceso servidor y el proceso auxiliar que genera la respuesta.

Common Gateway Interface

NCSA '93 - RFC 3875

- Una interfaz simple para ejecutar programas externos en un servidor web, en forma independiente de la plataforma (y del lenguaje de programación). El servidor actúa como una pasarela de aplicación: recibe la petición del cliente, selecciona el script CGI que manejará la petición, convierte el pedido del cliente a un pedido CGI, ejecuta el script, reconvierte la respuesta CGI y la remite al cliente. CGI define un conjunto de parámetros abstractos que describen la petición del cliente.
- El servidor es responsable de manejar la conexión y transferir los datos, mientras que el script maneja las cuestiones de aplicación, el mantenimiento de sesión y el procesamiento de los documentos.

Common Gateway Interface

- Dirección URL
 - *<scheme>* "://" *<server-name>* ":" *<server-port>* *<script-path>* *<extra-path>* "?" *<query-string>*
- Ejemplo
 - *http://www.tyr.unlu.edu.ar/cms/blog?page=7*
- Variables CGI relevantes
 - PATH_INFO, PATH_TRANSLATED, QUERY_STRING
 - REMOTE_ADDR, REMOTE_HOST, REMOTE_USER
 - REQUEST_METHOD, SCRIPT_NAME
 - SERVER_NAME, SERVER_PROTOCOL, SERVER_SOFTWARE

Método HTTP GET

- Los datos son enviados por el cliente al servidor codificados en la URI, después de "?".
- Los datos son enviados por el servidor al proceso auxiliar en la variable de entorno QUERY_STRING.
- Ejemplo: *http://www.example.com/catalogo.cgi?categoria=23&marca=Acme*
 - Variables de entorno (parcial):
HTTP_HOST = *www.example.com*
REQUEST_METHOD = GET
REQUEST_URI = */catalogo.cgi*
QUERY_STRING = *categoria=23&marca=Acme*

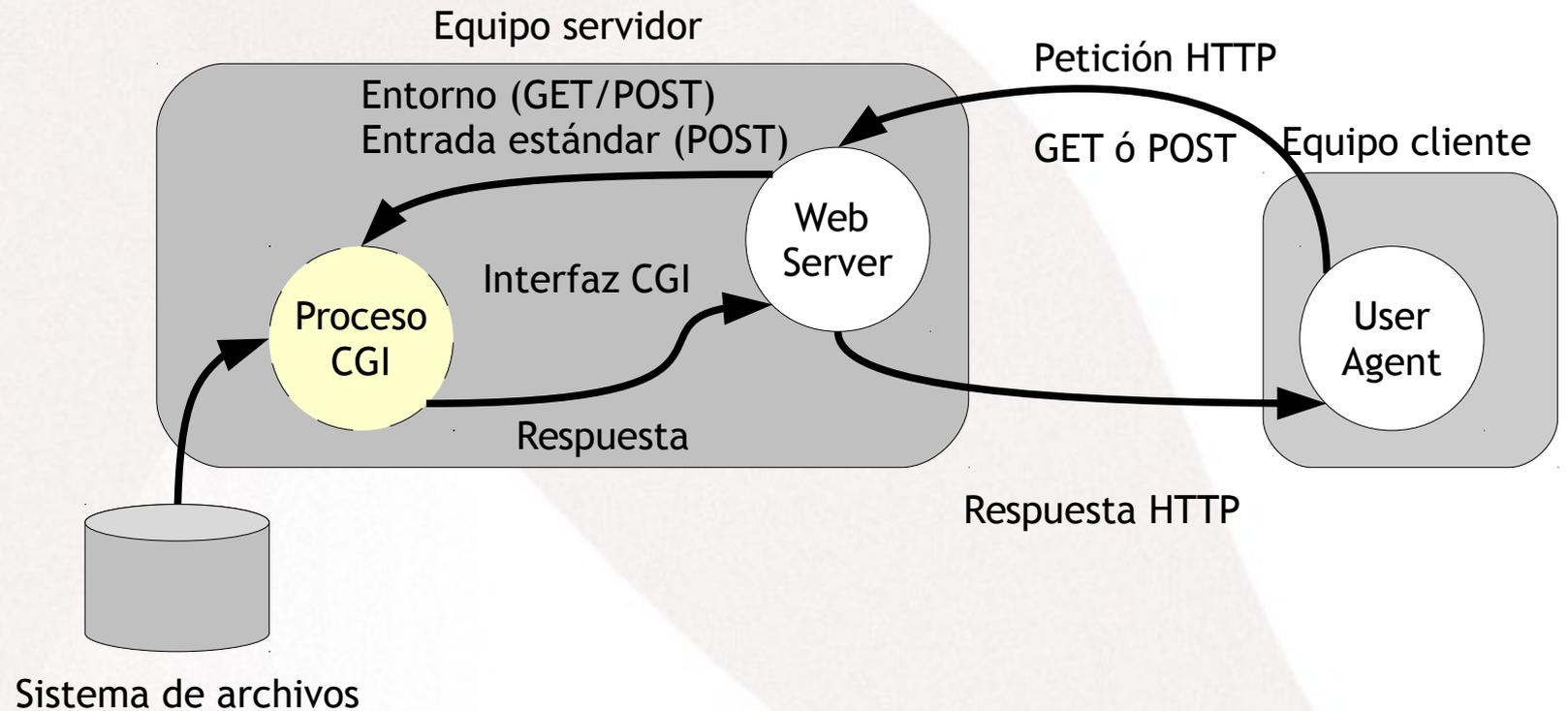
Método HTTP POST

- Los datos son enviados por el cliente al servidor en el cuerpo de la petición HTTP.
- Los datos son enviados por el servidor al proceso auxiliar como entrada estándar (stdin).
- Ejemplo: *http://www.example.com/catalogo.cgi*
 - Variables de entorno (parcial):
HTTP_HOST = *www.example.com*
REQUEST_METHOD = POST
REQUEST_URI = */catalogo.cgi*
QUERY_STRING =
 - Entrada estándar:
categoria=23
marca=Acme

Common Gateway Interface

- El proceso auxiliar puede estar desarrollado utilizando cualquier lenguaje que implemente la interfaz CGI: C, Bash, Perl, PHP, Python, etc...
- Todo script DEBE emitir una respuesta.
 - Status
 - Content-Type
 - Location
 - Body

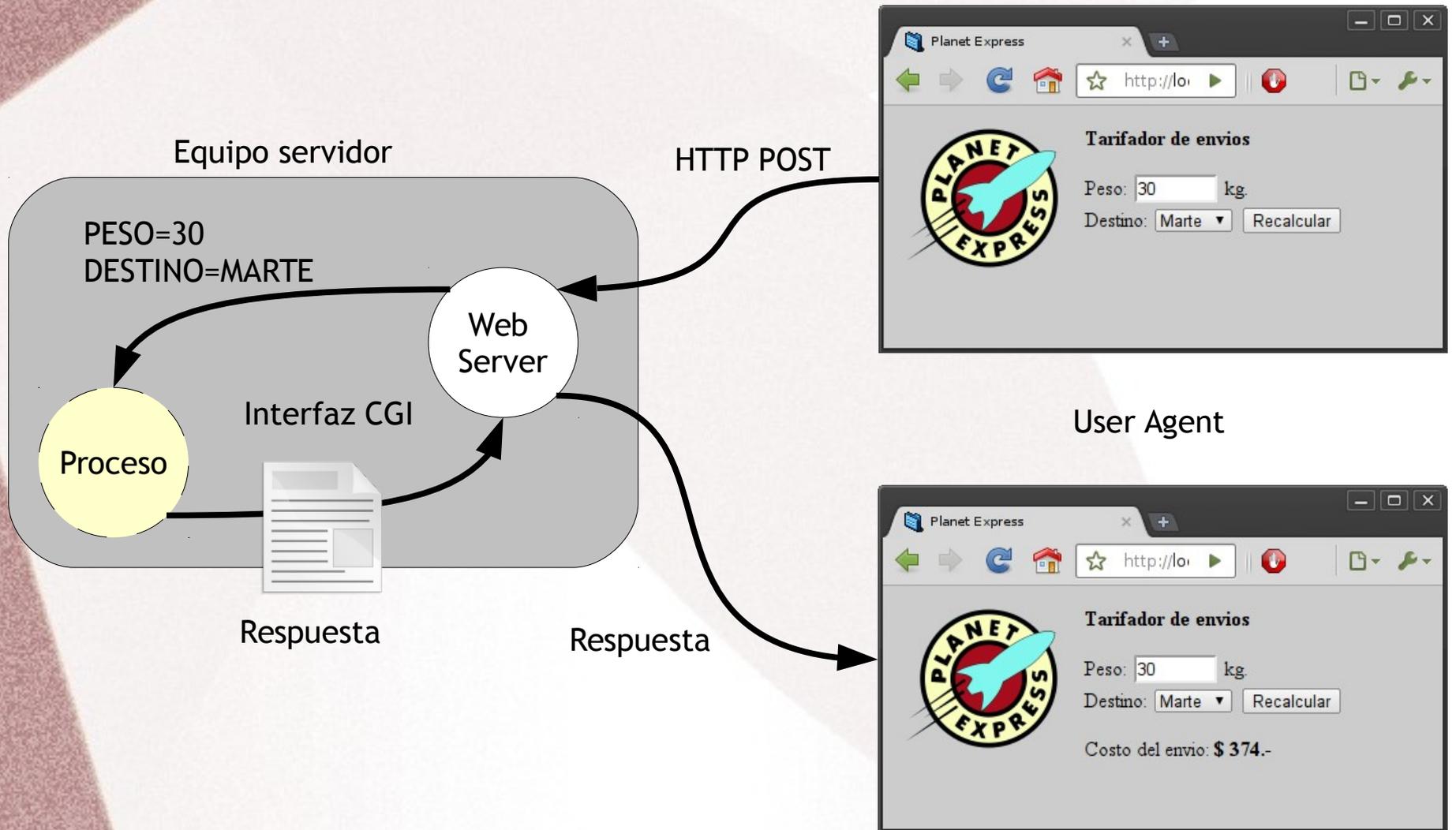
Common Gateway Interface



Ejemplo de código PHP

```
<?php
    $peso = $_POST['peso'];
    $destino = $_POST['destino'];
    $costo = $costo_base + costo_a($destino) * $peso;
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
    <title>Planet Express</title>
</head>
<body bgcolor="#CCCCCC">
    
    <form method="POST">
        <p><strong>Tarifador de envios</strong></p>
        <p>Peso: <input type="text" name="peso" value="30" size="5"> kg.<br>
        Destino: <select name="destino">
            <option>Marte</option>
            <option>Tierra</option>
            <option>Jupiter</option>
        </select>
        <input type="submit" name="submit" value="Recalcular"></p>
        <p>Costo del envio: <strong>$ <?php echo $costo ?></strong></p>
    </form>
</body>
</html>
```

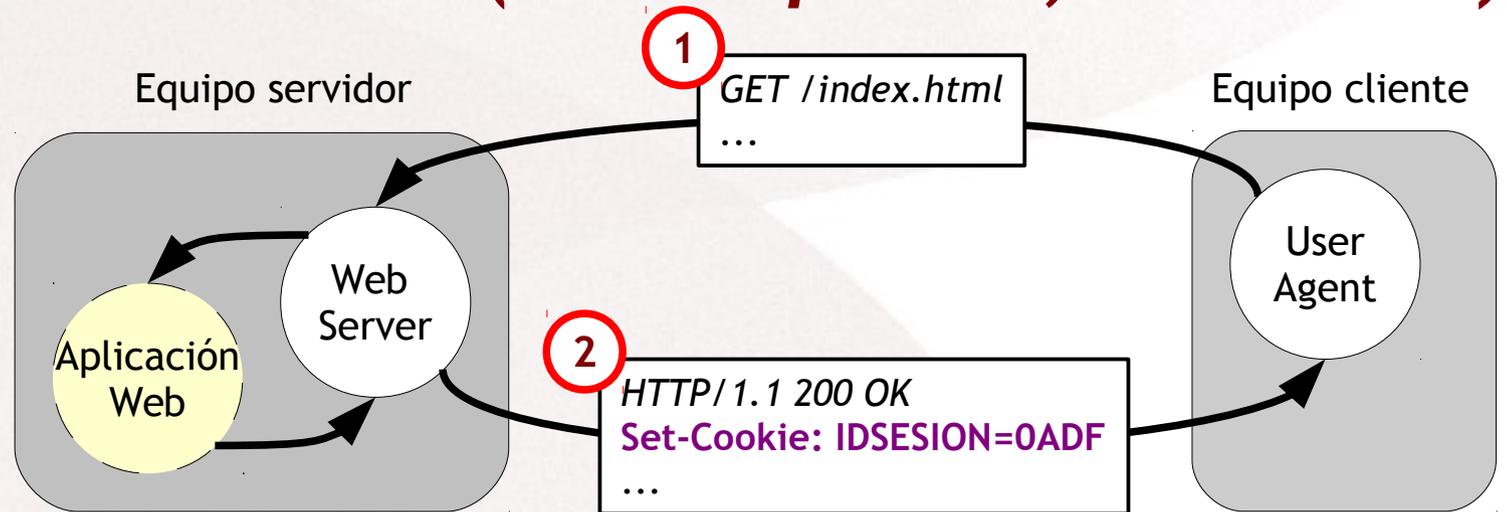
Common Gateway Interface



Mantenimiento de información de estado

- El protocolo HTTP **no mantiene información de estado** sobre el usuario entre transacciones sucesivas (se dice que es **stateless**).
- Muchas aplicaciones web requieren mantener esta información entre las diferentes peticiones HTTP para brindarle al usuario contenido personalizado (ej: datos de login, ubicación geográfica, etc.)
- Una solución adoptada por muchas aplicaciones web es el uso de **Cookies**, un fragmento de texto que el servidor web puede almacenar en el cliente y que el cliente web remite en el *Header* de cada petición.

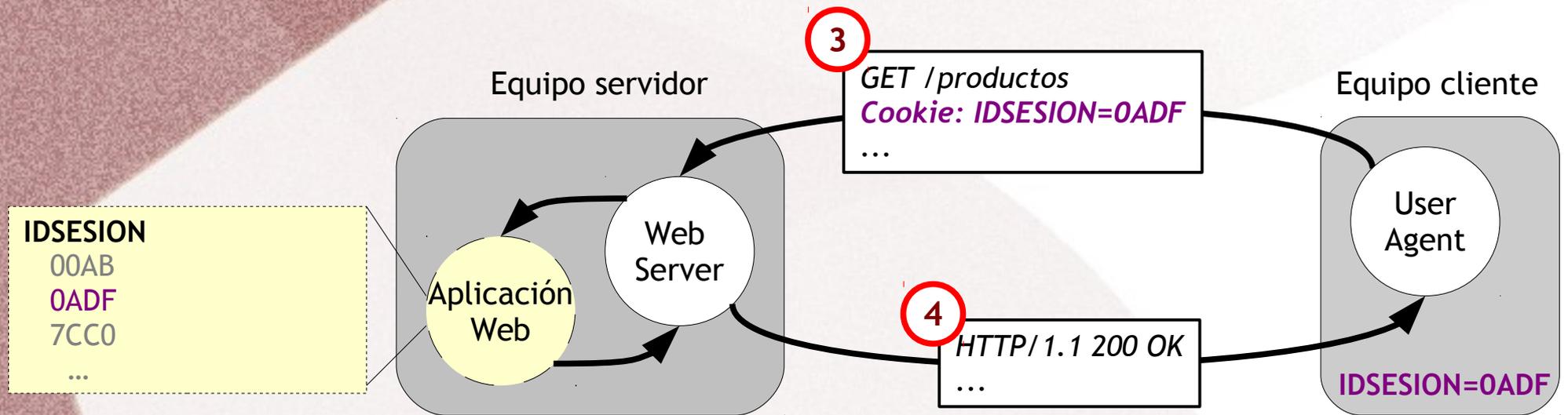
Cookies (Netscape '94, RFC 2965)



- El navegador web realiza una petición por un recurso al servidor web.
- El servidor web responde retornando el recurso solicitado e incluyendo con el encabezado *Set-Cookie* un par (*clave, valor*) que identificará al cliente.



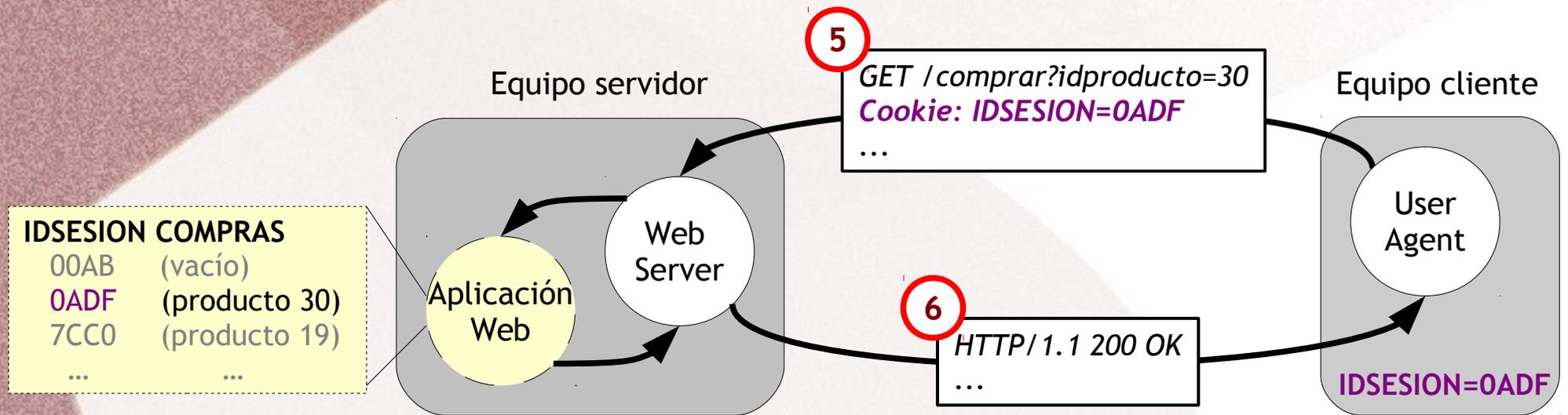
Cookies



- El cliente almacena los valores recibidos y en cada petición posterior al mismo servidor, incluirá un encabezado *Cookie* con los valores asignados.
- La aplicación web utiliza esos valores para identificar unívocamente a un cliente o sesión de usuario.



Cookies



- De esta forma, mientras el usuario interactúa con la aplicación web a través de peticiones y respuestas entre su navegador y el sitio remoto, la aplicación web puede mantener la información de estado de los diferentes usuarios y sesiones que requiera para su funcionamiento.



Cookies

Algunas cuestiones

- *De Privacidad*

Es posible asociar un usuario final con el historial de navegación dentro de un sitio web. Además, las empresas de publicidad en línea las utilizan para determinar a qué sitios accede cada navegante y crear así un perfil de intereses.

- *Técnicas*

Espacio de almacenamiento en el navegador, restricciones de dominio, expiración inconsistente, transmisión en texto plano, captura de cookies ajenas y posibilidad de falsificación.



HTML, Javascript, CSS (DHTML) (Netscape '96, W3C '96)

- **JavaScript** es un lenguaje liviano de scripting orientado a objetos e independiente de la plataforma diseñado para ser embebido en (extender) otros productos y aplicaciones, tal como un navegador web.
- Dentro de la aplicación que lo contiene, JavaScript puede interactuar y controlar los objetos del entorno mediante un modelo de documento (DOM).

HTML, Javascript, CSS (DHTML) *(Netscape '96, W3C '96)*

- ***Cascading Style Sheets (CSS)***, hojas de estilo en cascadas, no son un lenguaje de programación sino un lenguaje para describir el diseño y la presentación (fuentes, colores, posicionamiento de objetos) de un documento redactado en un lenguaje de marcado tal como HTML.
- CSS separa el contenido de la presentación, permitiendo diferentes presentaciones (en pantalla, impresión, disp. móviles) para un mismo contenido.

HTML 5 (WHATGW '04, W3C '14)

- Objetivo fundamental

“Proveer una forma estándar de que las aplicaciones web se comporten de manera similar a las aplicaciones nativas sin utilizar plugins en el navegador” -Ryan Gerry
- Comprende una colección de características individuales e independientes.
- Parte de esas características ya están soportadas en los navegadores actuales (móviles, por ej.).
- Es compatible hacia atrás con el HTML tradicional.
- Es sencillo respecto de otras alternativas de diseño.



HTML 5

Características

- Nuevos elementos semánticos (etiquetas).
 - Etiquetas obsoletas.
 - API para Scripting (canvas, drag&drop).
 - Audio y Video nativo (H.264, WebM, Ogg)
 - Aplicaciones fuera de línea.
 - Mejores controles en formularios.
 - API para historia de navegación.
 - Manejo de Errores.
 - Canvas 2D + WebGL (3D)
 - Microdata.
 - Cross-document messaging.
- especificaciones separadas

