Arquitecturas Distribuidas

Tema 4. II - Cookies

II. *Cookies*

1. Necesidad de mantener información de estado y HTTP
2. Sesiones
3. ¿Qué son las *cookies*?
4. Funcionamiento de *cookies*
5. Envío de *cookies* al cliente
6. Gestión de *cookies* en el cliente
7. Devolución de *cookies* al servidor
8. Ejemplos de transacciones con *cookies*

Necesidad de mantener información de estado y HTTP

- El protocolo HTTP es un protocolo de **petición-respuesta**
- Los servidores HTTP responden a cada cliente sin relacionar las peticiones con otras anteriores, o futuras.
- Sin embargo, es conveniente (para poder hacer ciertas aplicaciones) establecer "sesiones" que agrupen varias solicitudes/respuestas HTTP en un contexto mayor.
- El problema es que el protocolo HTTP no mantiene información de estado: cada petición es independiente de la anterior.

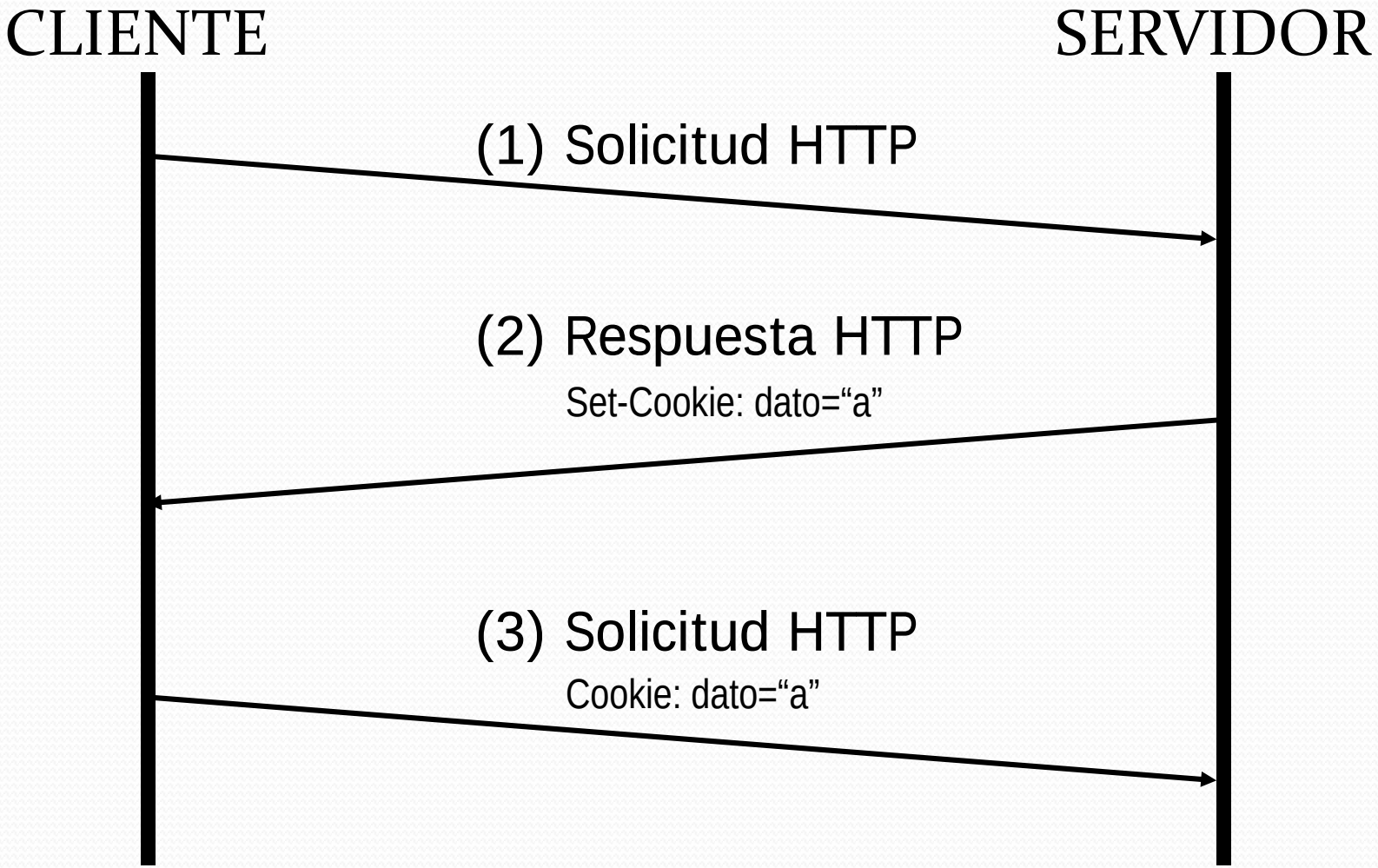
Sesiones

- Características de las sesiones:
  - Agrupan un conjunto de peticiones-respuestas
  - La sesión está implícita en el intercambio de información de estado
  - Tienen un comienzo y un final.
  - Un tiempo de vida relativamente pequeño.
  - La sesiones puede ser finalizadas por el servidor o por el agente de usuario (navegador).
- Ej. aplicación: "Carrito de la compra", donde el usuario elige en distintas peticiones distintos artículos para comprar.

¿Qué son *cookies*?

- *Cookies* es una técnica que permite a los servidores enviar información de estado al cliente, y recuperar dicha información.
- Consiste en que el cliente almacene una variable y su valor correspondiente y se lo devuelva al servidor en posteriores peticiones
- Básicamente pueden considerarse identificadores de sesión o estado
- Para ello, se utilizan dos nuevas cabeceras en HTTP:
  - En las respuestas: **Set-Cookie**.
  - En las solicitudes: **Cookie**.

Funcionamiento de *cookies*



Envío de *cookies* al cliente

- Se realiza a través de la cabecera de HTTP

Set-Cookie: "Set-cookie: paso=1;Max-Age=1102"

```
set-cookie = "Set-Cookie:" cookies
cookies = 1#cookie
cookie = NAME "=" VALUE *(";" cookie-modificador)
NAME = attr
VALUE = value
cookie-modificador = "Comment" "=" value
            | "Domain" "=" value
            | "Max-Age" "=" value
            | "Path" "=" value
            | "Secure"
            | "Version" "=" 1*DIGIT
```

Envío de *cookies* al cliente

- Es decir, Set-Cookie contiene un par NAME=VALUE, que puede contener modificadores separados por ";".
- Los NAME que comienzan por $ están reservados.
- VALUE puede ser cualquier información que desee el servidor, su contenido es opaco al cliente.
- Modificadores:
  - `Comment:` el servidor informa del uso de la *cookie*.
  - `Domain`: indica el dominio en el que la *cookie* es válida.
  - `Max-Age`: validez de la *cookie* en segundos, después se descarta.
  - `Path`: indica el subconjunto de URL a las que afecta la *cookie*.
  - `Secure`: indica que el cliente debe usar un entorno seguro para contactar con el servidor (HTTP sobre SSL o similar).
  - `Versión`: la versión de la especificación de mantenimiento de estado que es.

Gestión de *cookies* en el cliente

- Interpretación de la cabecera Set-Cookie por el agente de usuario:
  - La información de estado se almacena separada según el servidor de origen = se agrupan las cookies de cada servidor (identificado por el nombre de dominio)
  - Si algún modificador no se especifica, se asume:
    - `Version` → "Old cookies", la definición original de Netscape.
    - `Domain` → Host al que se ha efectuado la solicitud.
    - `Max-Age` → Descartar la solicitud al finalizar la ejecución del agente de usuario.
    - `Path` → La URL de la última petición, hasta el "/" más a la derecha.
    - `Secure` → No necesaria, es decir, la *cookie* puede ser enviada a través de un canal inseguro.

Gestión de *cookies* en el cliente

- Una *cookie* puede ser descartada en caso de:
  - No ser admitidas por el usuario.
  - No verificarse ciertas condiciones intrínsecas de seguridad en los modificadores.
- Si se recibe una *cookie* con el mismo NAME al de una pre-existente, y el `Domain` y `Path` coinciden exactamente en ambas, la nueva *cookie* sustituye a la anterior. Si la nueva cookie tiene `Max-Age` igual a cero, la nueva y la anterior son descartadas (el servidor usa este método para cerrar la sesión).

Gestión de *cookies* en el cliente

- El número de *cookies* almacenables es finito, por tanto, el agente de usuario debe eliminar alguna *cookie* cuando ya no queda espacio para una nueva (generalmente, se usa un algoritmo tipo "descartar la usada hace más tiempo").
- Cuando vence la duración de una *cookie*, ésta se descarta. El navegador la elimina.

Devolución de *cookies* al servidor

- Se realiza a través de la cabecera de HTTP **Cookie**: "Cookie: paso=1"

```
cookie = "Cookie:" cookie-version
           1*((";" | ",") cookie-value)
cookie-value = NAME "=" VALUE [";" path] [";" domain]
cookie-version = "$Version" "=" value
NAME = attr
VALUE = value
path = "$Path" "=" value
domain = "$Domain" "=" value
```

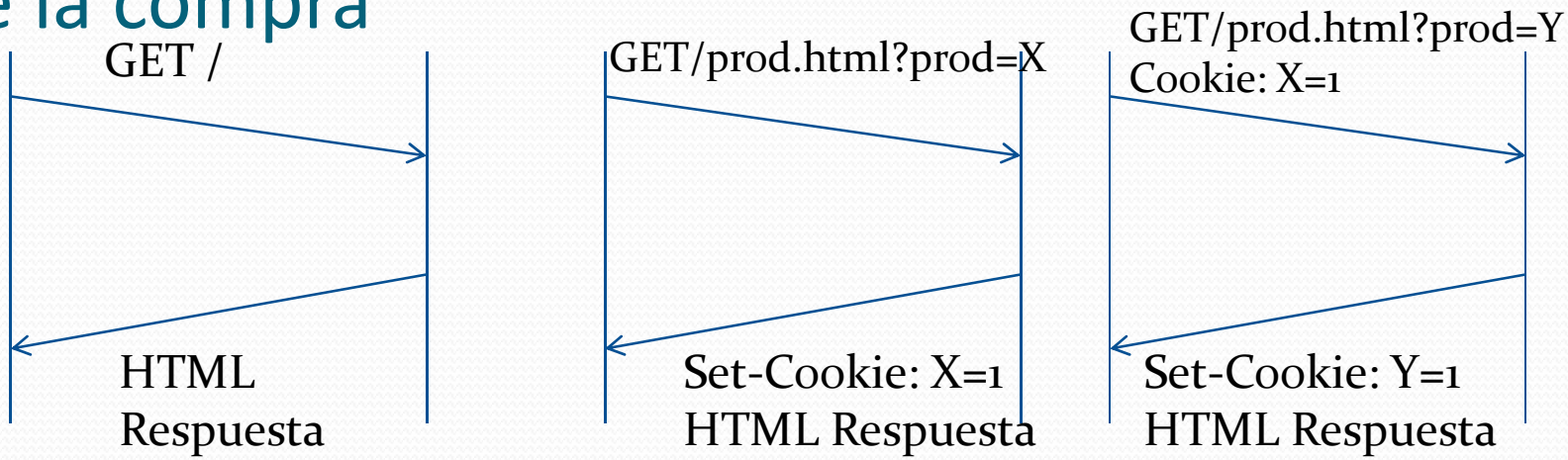
Devolución de *cookies* al servidor

- Es decir, **Cookie** contiene una colección de pares NAME=VALUE (con posibles modificadores separados por “;” ó “,”).
- Las *cookies* seleccionadas para ser devueltas son aquellas que verifican:
  - El nombre canónico del servidor al que se envía coincide con el almacenado en `Domain`.
  - El atributo `Path` de la *cookie* coincide **exactamente** con un prefijo del URL que se quiere solicitar (hasta la última barra a la derecha)
  - La cookie no ha expirado (atributo `Max-Age`).

Devolución de *cookies* al servidor

- Si más de una *cookie* cumple estos criterios, se envían ordenadas, desde la que contiene el `Path` más específico, a la que menos.

Ejemplos de transacciones con *cookies:* "Carrito de la compra"



Usuario solicita el catálogo

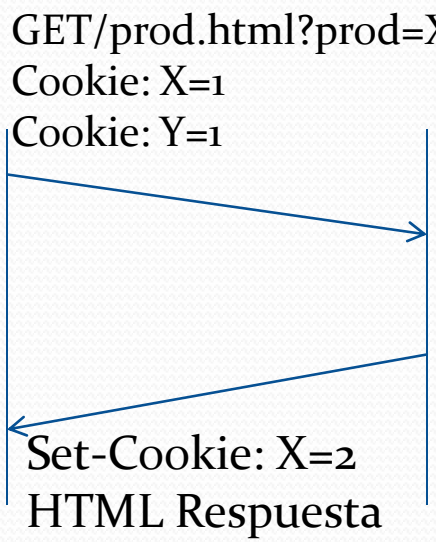
- Servidor devuelve el catálogo (HTML)

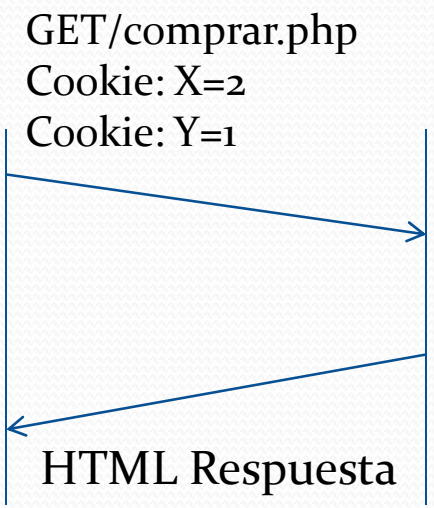
Usuario selecciona producto X

- Servidor devuelve una cookie X=1
- Se vuelve a devolver el catálogo

Usuario selecciona el producto Y

- Servidor recibe la cookie X=1
- Servidor devuelve una cookie Y=1
- Se vuelve a devolver el catálogo





Usuario selecciona de nuevo X

- Servidor recibe las cookies anteriores, X=1 e Y=1
- Servidor actualiza el número de productos X con la cookie X=2, se sobreescribe en el cliente el valor de la cookie X

Usuario decide comprar

- Servidor recibe las cookies anteriores
- Con el valor de las cookies sabe que el usuario quiere comprar dos unidades de X y una de Y
- El servidor devuelve el importe y el formulario para introducir el nº de tarjeta de crédito

Privacidad y cookies

- Cookies de “terceras partes”
  - ¿puede un servidor insertar cookies con un dominio diferente al suyo?
  - Pero, ¿qué ocurre si un documento web contiene una imagen alojada en otro servidor?
- En principio, al descargar una imagen de un servidor externo se descargarían las cookies de dicho servidor
- Además, el cliente devolvería cookies previas al realizar la petición
- Dichas cookies permiten crear un perfil anónimo de páginas visitadas
- La mayoría de los navegadores permiten bloquear las cookies de terceras partes

Referencias y bibliografía

- *Cookies*
  - RFC 2109: "HTTP State Management Mechanism", disponible en:
    - http://www.faqs.org/rfcs/rfc2109.html
    - http://ait.upct.es/asignaturas/ad/manuales/COOKIES/RFC%202109.html