# Cookie

Una ***cookie*** (pronunciado ['ku.ki]; literalmente *galleta*) es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas. En ocasiones también se le llama "huella".

Las inventó Lou Montulli, un antiguo empleado de Netscape Communications. Al ser el protocolo HTTP incapaz de mantener información por sí mismo, para que se pueda conservar información entre una página vista y otra (como *login* de usuario, preferencias de colores, etc), ésta debe ser almacenada, ya sea en la URL de la página, en el propio servidor, o en una *cookie* en el ordenador del visitante.

De esta forma, los usos más frecuentes de las *cookies* son:

- Llevar el control de usuarios: cuando un usuario introduce su nombre de usuario y contraseña, se almacena una *cookie* para que no tenga que estar introduciéndolas para cada página del servidor. Sin embargo una cookie no identifica a una persona, sino a una combinación de computador y navegador.
- Conseguir información sobre los hábitos de navegación del usuario, e intentos de spyware, por parte de agencias de publicidad y otros. Esto puede causar problemas de privacidad y es una de las razones por la que las *cookies* tienen sus detractores.

Originalmente, sólo podían ser almacenadas por petición de un CGI desde el servidor, pero Netscape dio a su lenguaje Javascript la capacidad de introducirlas directamente desde el cliente, sin necesidad de CGIs. En un principio, debido a errores del navegador, esto dio algunos problemas de seguridad. Estas vulnerabilidades fueron descubiertas por Roberto Santizo.[*cita requerida*] Las cookies pueden ser borradas, aceptadas o bloqueadas según desee, para esto sólo debe configurar convenientemente el navegador web.

## Propósito

Las *cookies* son utilizadas habitualmente por los servidores web para diferenciar usuarios y para actuar de diferente forma dependiendo del usuario. Las *cookies* se inventaron para ser utilizadas en una cesta de la compra virtual, que actúa como dispositivo virtual en el que el usuario va "colocando" los elementos que desea adquirir, de forma que los usuarios pueden navegar por el sitio donde se muestran los objetos a la venta y añadirlos y eliminarlos de la cesta de la compra en cualquier momento. Las *cookies* permiten que el contenido de la cesta de la compra dependa de las acciones del usuario.

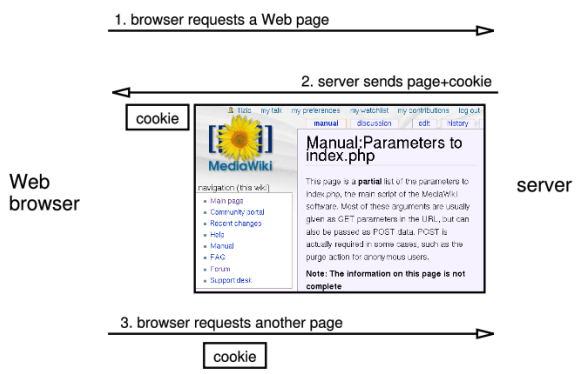
Si bien las cookies pretenden facilitar el acceso a las páginas web visitadas con anterioridad, no tienen acción sobre las actividades multimedia, ya sea video, audio o imágenes de caché considerable, dado que, como son almacenadas en la memoria temporal del disco duro, ocuparían demasiado espacio. Sus funciones se limitan únicamente a almacenar los ficheros HTML para facilitar el acceso a ellos por parte del usuario.

Otro uso de las *cookies* es identificarse en un sitio web. Los usuarios normalmente se identifican introduciendo sus credenciales en una página de validación; las *cookies* permiten al servidor saber que el usuario ya está validado, y por lo tanto se le puede permitir acceder a servicios o realizar operaciones que están restringidas a usuarios no identificados.

Otros sitos web utilizan las *cookies* para personalizar su aspecto según las preferencias del usuario. Los sitios que requieren identificación a menudo ofrecen esta característica, aunque también está presente en otros que no la requieren. La personalización incluye tanto presentación como funcionalidad. Por ejemplo, las páginas de Wikipedia permiten a los usuarios identificados elegir un estilo de presentación a su gusto; el motor de búsqueda de Google permite a los usuarios (incluso a los no registrados) decidir cuántos resultados de búsqueda quieren ver en cada página.

Las Cookies se utilizan también para realizar seguimientos de usuarios a lo largo de un sitio web. Las *cookies* de terceros y los errores en servidores web que se explican más abajo también permiten el seguimiento entre diferentes sitios. El seguimiento en un mismo sitio normalmente se hace con la intención de mantener estadísticas de uso, mientras que el seguimiento entre sitios normalmente se orienta a la creación de perfiles de usuarios anónimos por parte de las compañías de publicidad, que luego se usarán para orientar campañas publicitarias (decidir qué tipo de publicidad utilizar) basadas en perfiles de usuarios.

## Realización




Una posible interacción entre un navegador web y un servidor, en la que el servidor envía una *cookie* al navegador y el navegador la devuelve cuando solicita otra página.

Técnicamente, las *cookies* son trozos de datos arbitrarios definidos por el servidor web y enviados al navegador. El navegador los devuelve sin modificar al servidor, reflejando así un *estado* (memoria de eventos anteriores) en las transacciones HTTP, que de otra manera serían independientes de estado. Sin las *cookies*, cada petición de una página web o un componente de una página web sería un evento aislado, sin ninguna relación con el resto de peticiones de otras páginas del mismo sitio. Pero devolviendo una *cookie* al servidor web, el navegador le proporciona un medio para relacionar la solicitud de la página actual con solicitudes de páginas anteriores. Además de ser definidas por un servidor web, las *cookies* también pueden ser definidas por un script en un lenguaje como JavaScript, si éste está soportado y habilitado en el navegador web.

Las especificaciones de *cookies*[1] [2] sugieren que los navegadores deben soportar un número mínimo de *cookies* o una cantidad mínima de memoria para almacenarlas. En concreto, se espera que un navegador sea capaz de almacenar al menos 300 cookies de 4 kilobytes cada una y al menos 20 *cookies* por servidor o dominio.

El servidor que establece la *cookie* puede especificar una fecha de borrado, en cuyo caso la *cookie* será borrada en esa fecha. Un sitio de compras podría querer ayudar a clientes potenciales recordando las cosas que había en su cesta de la compra, incluso si cierran el navegador sin realizar la compra y vuelven más tarde, para evitar que tengan que buscar los productos de nuevo. En ese caso, el servidor crearía una *cookie* con fecha de borrado según el deseo del diseñador del sitio web. Si no se define una fecha de borrado, la *cookie*

es borrada cuando el usuario cierra su navegador. Por lo tanto, definir una fecha de borrado es una manera de hacer que la *cookie* sobreviva entre sesiones. Por esta razón, las *cookies* con fecha de borrado se llaman *persistentes*.

## Ideas equivocadas

Desde su introducción en Internet han circulado ideas equivocadas acerca de las *cookies*.[3] [4] En 2005 Jupiter Research publicó los resultados de un estudio,[5] según el cual un importante porcentaje de entrevistados creían cierta alguna de las siguientes afirmaciones:

- Las *cookies* son similares a gusanos y virus en que pueden borrar datos de los discos duros de los usuarios.
- Las *cookies* son un tipo de spyware porque pueden leer información personal almacenada en el ordenador de los usuarios.
- Las *cookies* generan popups.
- Las *cookies* se utilizan para generar spam.
- Las *cookies* sólo se utilizan con fines publicitarios.

En realidad, las *cookies* son sólo datos, no código, luego no pueden borrar ni leer información del ordenador de los usuarios.[6] Sin embargo, las cookies permiten detectar las páginas visitadas por un usuario en un sitio determinado o conjunto de sitios. Esta información puede ser recopilada en un *perfil* de usuario. Estos perfiles son habitualmente anónimos, es decir, no contienen información personal del usuario (nombre, dirección, etc). De hecho, no pueden contenerla a menos que el propio usuario la haya comunicado a alguno de los sitios visitados. Pero aunque anónimos, estos perfiles han sido objeto de algunas preocupaciones relativas a la privacidad.

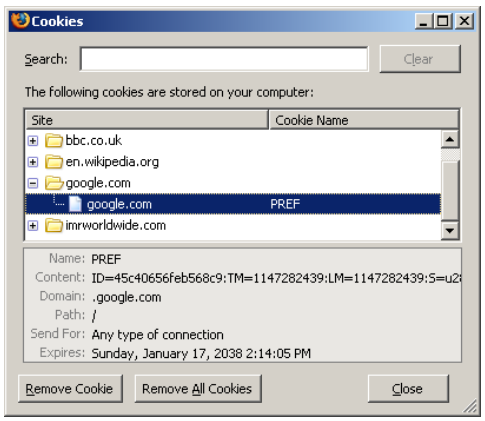
Según el mismo informe, un gran porcentaje de los usuarios de Internet no saben cómo borrar las *cookies*.

## Configuración del navegador

La mayor parte de los navegadores modernos soportan las *cookies*. Sin embargo, un usuario puede normalmente elegir si las *cookies* deberían ser utilizadas o no. A continuación, las opciones más comunes:[7]

1. Las *cookies* no se aceptan nunca.
2. El navegador pregunta al usuario si se debe aceptar cada *cookie*.
3. Las *cookies* se aceptan siempre.

El navegador también puede incluir la posibilidad de especificar mejor qué *cookies* tienen que ser aceptadas y cuáles no. En concreto, el usuario puede normalmente aceptar alguna de las siguientes opciones: rechazar las *cookies* de determinados dominios; rechazar las *cookies* de terceros (ver más abajo); aceptar *cookies* como no persistentes (se eliminan cuando el navegador se cierra); permitir al servidor crear *cookies* para un dominio diferente. Además, los navegadores pueden también permitir a los usuarios ver y borrar *cookies* individualmente.



El gestor de *cookies* de Firefox, mostrando los detalles de varias *cookies* por dominio

La mayoría de los navegadores que soportan JavaScript permiten a los usuarios ver las *cookies* que están activas en una determinada página escribiendo `javascript:alert("Cookies: "+document.cookie)` en el campo de dirección.

La especificación P3P incluye la posibilidad de que un servidor defina una política de privacidad que especifique qué tipo de información recoge y con qué propósito. Estas políticas incluyen (entre otras cosas) el uso de información recopilada a través de *cookies*. Según la especificación P3P, un navegador puede aceptar o rechazar *cookies* comparando la política de privacidad con las preferencias del usuario almacenadas, o preguntar al usuario, ofreciéndole la política de privacidad declarada por el servidor.

## Privacidad y cookies de terceros

Las *cookies* tienen implicaciones importantes en la privacidad y el anonimato de los usuarios de la web. Aunque las *cookies* sólo se envían al servidor que las definió o a otro en el mismo dominio, una página web puede contener imágenes y otros componentes almacenados en servidores de otros dominios. Las *cookies* que se crean durante las peticiones de estos componentes se llaman *cookies de terceros*.

Las compañías publicitarias utilizan *cookies* de terceros para realizar un seguimiento de los usuarios a través de múltiples sitios. En concreto, una compañía publicitaria puede seguir a un usuario a través de todas las páginas donde ha colocado imágenes publicitarias o *web bugs*. El conocimiento de las páginas visitadas por un usuario permite a estas compañías dirigir su publicidad según las supuestas preferencias del usuario.

La posibilidad de crear un perfil de los usuarios se ha considerado como una potencial amenaza a la privacidad, incluso cuando el seguimiento se limita a un solo dominio, pero especialmente cuando es a través de múltiples dominios mediante el uso de *cookies* de terceros. Por esa razón, algunos países tienen legislación sobre *cookies*.

El gobierno de los Estados Unidos definió estrictas reglas para la creación de *cookies* en el año 2000, después de que se conociese que la Oficina de Control de Drogas Nacional de la Casa Blanca utilizaba *cookies* para seguir a los usuarios que tras visitar su campaña anti-drogas, visitaban sitios relacionados con la fabricación o el uso de drogas. En 2002, el activista por la privacidad Daniel Brandt averiguó que la CIA había estado definiendo *cookies* persistentes en ordenadores durante diez años. Cuando les informó de que estaban violando la política, la CIA confirmó que esas *cookies* no habían sido creadas

intencionadamente, y dejó de utilizarlas.[8] El 25 de diciembre de 2005, Brandt descubrió que la Agencia de Seguridad Nacional había estado creando dos *cookies* persistentes en los ordenadores de sus visitantes debido a una actualización de software. Tras ser informada, la agencia deshabilitó inmediatamente las *cookies*.[9]

La directiva de la Unión Europea de 2002 sobre privacidad en las telecomunicaciones [10] contiene reglas sobre el uso de *cookies*. En concreto, en el artículo 5, párrafo 3 establece que el almacenamiento de datos (como *cookies*) en el ordenador de un usuario sólo puede hacerse si: 1) el usuario recibe información sobre cómo se utilizan esos datos; y 2) el usuario tiene la posibilidad de rechazar esa operación. Sin embargo, este artículo también establece que almacenar datos que son necesarios por motivos técnicos está permitido como excepción. Se esperaba que esta directiva hubiese comenzado su aplicación desde octubre de 2003, pero un informe de diciembre de 2004 [11] dice (página 38) que no ha sido aplicado en la práctica, y que algunos países miembros (Eslovaquia, Letonia, Grecia, Bélgica y Luxemburgo) ni siquiera la han transpuesto a su legislación. El mismo informe sugiere un profundo análisis de la situación en los estados miembros.

## Inconvenientes de las *cookies*

Además de lo relativo a la privacidad que ya se ha mencionado, hay otras razones por las que el uso de *cookies* ha recibido cierta oposición: no siempre identifican correctamente a los usuarios, y se pueden utilizar para ataques de seguridad.

### Identificación inexacta

Si se utiliza más de un navegador en un ordenador, cada uno tiene su propio almacenamiento de *cookies*. Por lo tanto, las *cookies* no identifican a una persona, sino a una combinación de cuenta de usuario, ordenador y navegador. De esta manera, cualquiera que utilice varias cuentas, varios ordenadores, o varios navegadores, tiene también múltiples conjuntos de cookies.

De la misma manera, las cookies no diferencian entre varias personas que utilicen el mismo ordenador o navegador, si éstos no utilizan diferentes cuentas de usuario.

### Robo de *cookies*

Durante el funcionamiento normal, las *cookies* se envían de un extremo al otro entre el servidor (o grupo de servidores en el mismo dominio) y el ordenador del usuario que está navegando. Dado que las *cookies* pueden contener información sensible (nombre de usuario, un testigo utilizado como autenticación, etc.), sus valores no deberían ser accesibles desde otros ordenadores. Sin embargo, las *cookies* enviadas sobre sesiones HTTP normales son visibles a todos los usuarios que pueden escuchar en la red utilizando un *sniffer* de paquetes. Estas *cookies* no deben contener por lo tanto información sensible. Este problema se puede solventar mediante el uso de https, que invoca seguridad de la capa de transporte para cifrar la conexión.

El scripting entre sitios permite que el valor de las *cookies* se envíe a servidores que normalmente no recibirían esa información. Los navegadores modernos permiten la ejecución de segmentos de código recibidos del servidor. Si las *cookies* están accesibles durante la ejecución, su valor puede ser comunicado de alguna manera a servidores que no deberían acceder a ellas. El proceso que permite a una parte no autorizada recibir una *cookie* se llama *robo de cookies*, y el cifrado no sirve contra este tipo de ataque.[12]

Esta posibilidad es explotada normalmente por atacantes de sitios que permiten a los usuarios el envío de contenido HTML. Introduciendo un segmento de código adecuado en un envío HTML, un atacante puede recibir las *cookies* de otros usuarios. El conocimiento de estas *cookies* puede después ser explotado mediante la conexión a los sitios en los que se utilizan las *cookies* robadas, siendo así identificado como el usuario a quien se le robaron las *cookies*

### Falsificación de *cookies*

Aunque las *cookies* deben ser almacenadas y enviadas de vuelta al servidor sin modificar, un atacante podría modificar el valor de las *cookies* antes de devolverlas. Si, por ejemplo, una *cookie* contiene el valor total de la compra de un usuario en un sitio web, cambiando ese valor el servidor podría permitir al atacante pagar menos de lo debido por su compra. El proceso de modificar el valor de las *cookies* se denomina *falsificación de cookies* y a menudo se realiza tras un *robo de cookies* para hacer un ataque persistente.

Sin embargo, la mayoría de los sitios web solo almacenan en la *cookie* un identificador de sesión —un número único utilizado para identificar la sesión del usuario— y el resto de la información se almacena en el propio servidor. En este caso, el problema de la falsificación de *cookies* queda prácticamente eliminado.

### *Cookies* entre sitios (*cross-site cooking*)

Cada sitio debe tener sus propias *cookies*, de forma que un sitio *malo.net* no tenga posibilidad de modificar o definir *cookies* de otro sitio como *bueno.net*. Las vulnerabilidades de *cross-site cooking* de los navegadores permiten a sitios maliciosos romper esta regla. Esto es similar a la falsificación de *cookies*, pero el atacante se aprovecha de usuarios no malintencionados con navegadores vulnerables, en vez de atacar el sitio web directamente. El objetivo de estos ataques puede ser realizar una fijación de sesión (robo de sesión en un sitio web).

## Alternativas a las *cookies*

Algunas de las operaciones que se pueden realizar mediante *cookies* también se pueden hacer mediante otros mecanismos. Sin embargo, estas alternativas a las *cookies* tienen sus propios inconvenientes, que acaban convirtiendo a las *cookies* en la opción preferida en la práctica. La mayoría de las alternativas descritas a continuación permiten el seguimiento del usuario, si bien es cierto que no de forma tan fiable. Es por ello que la privacidad sigue siendo un problema, incluso si el navegador rechaza las *cookies* y el servidor no las define.

### Dirección IP

Una técnica poco fiable de realizar un seguimiento de usuarios se basa en almacenar la dirección IP del ordenador que solicita las páginas. Esta técnica ha estado disponible desde los inicios de World Wide Web, al ser necesario para la descarga de páginas que el servidor que las tiene conozca la dirección IP del ordenador en el que corre el navegador, o de su servidor proxy si lo hay. El servidor puede guardar esta información, independientemente del uso o no de *cookies*.

Sin embargo, estas direcciones son normalmente menos fiables que las *cookies* para la identificación de un usuario, debido a que los ordenadores y *proxies* pueden estar compartidos por varios usuarios, y el mismo ordenador puede tener asignadas diferentes

direcciones IP en diferentes sesiones (caso típico en conexiones telefónicas, aunque también a través de ADSL y otras tecnologías). La fiabilidad de esta técnica se puede aumentar mediante el uso de otra característica del protocolo HTTP: cuando un navegador solicita una página porque el usuario ha seguido un link, la petición que se envía al servidor contiene la URL de la página donde el link estaba localizado. Si el servidor almacena esas URL, se puede rastrear el camino de páginas visitadas por el usuario de forma más precisa. Sin embargo, estos rastreos son menos fiables que los que proporcionan las *cookies*, ya que varios usuarios pueden acceder a la misma página desde el mismo ordenador, router con NAT o proxy, y después seguir links diferentes. Además, esta técnica sólo permite el rastreo, y no puede reemplazar a las *cookies* in sus otros usos.

El seguimiento de direcciones IP puede ser imposible en algunos sistemas que se utilizan precisamente para mantener el anonimato en Internet, tales como Tor. Con tales sistemas, no sólo puede un navegador utilizar varias direcciones a lo largo de una sesión, sino que varios usuarios podrían aparecer como si utilizasen la misma dirección IP, convirtiendo por lo tanto el uso de las direcciones IP en una técnica absolutamente inútil para el rastreo de usuarios.

### URL (query string)

Una técnica más precisa consiste en incrustar información en la URL. Normalmente se usa para este fin la query string que es parte de la URL, pero también se pueden utilizar otras partes. El mecanismo de sesión de PHP utiliza este método si las *cookies* no están habilitadas.

Este método consiste en que el servidor web añade *query strings* a los enlaces de la página web que contiene, a la hora de servirla al navegador. Cuando el usuario sigue un enlace, el navegador devuelve al servidor la *query string* añadidos a los enlaces.

Las *query strings* utilizadas de esta manera son muy similares a las *cookies*, siendo ambas porciones de información definidos por el servidor y devueltas por el navegador posteriormente. Sin embargo, existen diferencias: dado que una *query string* es parte de una URL, si la URL es reutilizada posteriormente, se estará enviando al servidor la misma porción de información. Si, por ejemplo, las preferencias de un usuario están codificadas en la *query string* de una URL, y el usuario envía esa URL a otro usuario por algún medio, esas preferencias serán utilizadas también por ese otro usuario.

Además, incluso si el mismo usuario accede a la misma página dos veces, no hay garantía de que se utilice la misma *query string* en las dos. Si, por ejemplo, el mismo usuario llega a la misma página dos veces, una proveniente de otra página del mismo servidor web, y otra de un buscador, las respectivas *query strings* serán normalmente diferentes, mientras que las *cookies* hubiesen sido idénticas. Para más detalles, véase query string.

Otras desventajas de las *query strings* están relacionadas con la seguridad: almacenar en una *query string* información que identifica una sesión permite o simplifica los ataques de fijación de sesión, ataques de seguimiento de referentes y otros exploits. La transferencia de identificadores de sesión en forma de *cookies* es más segura.

Otra desventaja de las *query strings* tiene que ver con la forma en que la web está diseñada. Las URL deberían apuntar a recursos y ser "opacas". Véase Representational State Transfer. Si se tiene una URL que incluye una *query string*, ya no es la ubicación real del recurso.

## Autenticación HTTP

Para la autenticación, el protocolo HTTP incluye mecanismos tales como el digest access authentication, que permite acceder a una página web sólo cuando el usuario ha facilitado un nombre de usuario y contraseña correctos. Una vez que se han introducido los credenciales, el navegador las almacena y las utiliza para acceder a las páginas siguientes, sin pedirlas de nuevo al usuario. Desde el punto de vista del usuario, el efecto es el mismo que si se usan *cookies*: el nombre de usuario y palabra clave sólo se piden una vez, y a partir de entonces el usuario obtiene acceso a las páginas del servidor. Internamente, el nombre de usuario y la contraseña se envían al servidor con cada petición del navegador. Esto quiere decir que alguien que estuviese escuchando este tráfico podría leer esta información y almacenarla para su uso posterior. Las sesiones, no obstante, normalmente expiran tras un periodo de inactividad determinado, quedando así invalidadas para, por ejemplo, recuperar la sesión que tenía el usuario que estaba navegando.

## Objetos Macromedia Flash almacenados localmente

Si un navegador incluye el plugin de Macromedia Flash Player, se puede utilizar la función *Local Shared Objects* del mismo, de una forma muy similar a las *cookies*. Los *Local Stored Objects* pueden ser una opción interesante para los desarrolladores web porque la mayoría de los usuarios de Windows tienen Flash Player instalado, el tamaño máximo por defecto de los objetos es 100 kb, y los controles de seguridad son distintos de los controles de usuario para las *cookies*, de forma que los *Local Shared Objects* pueden estar habilitados cuando las *cookies* no lo están.

## Persistencia en el cliente

Algunos navegadores web soportan un mecanismo de persistencia basado en script que permite que la página almacene información localmente para su uso posterior. Internet Explorer, por ejemplo, soporta información persistente en el historial del navegador, en los favoritos, en un almacenamiento XML, o directamente en una página web guardada en disco.[13]

### Propiedad *window.name de JavaScript*

Si está habilitado el uso de JavaScript, se puede utilizar la propiedad `window.name` del objeto `window` para almacenar información de forma persistente. Esta propiedad permanece inalterada durante la carga de otras páginas web. Este pequeño hack no es muy conocido, y por lo tanto no ha sido considerado un fallo de seguridad. Además, el uso de `window.name` tiene problemas de compatibilidad con navegadores, ya que que algunos, como los basados en Mozilla de los que Mozilla Firefox es un ejemplo, no soportan la persistencia con JavaScript utilizando `window.name`.[14]

# Implementación

## Creando una *cookie*

La transferencia de páginas Web sigue HTTP. A pesar de las cookies, exploradores piden una página de servidores para enviarles un texto corto llamado HTTP. Por ejemplo, para acceder a la página http://www.w3.org/index.html, los exploradores se conectan al servidor www.w3.org mandando una petición que se parece a la siguiente:

```
GET /index.html HTTP/1.1
```

**explorador** → **servidor**

El servidor responde al enviar la página pedida precedida por un texto similar, llamado encabezado HTTP. Este paquete puede contener líneas peticionando al explorador para que guarde cookies:

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value (content of page)
```

**explorador** ← **servidor**

La línea `Set-cookie` es solo enviada si el servidor desea que el explorador guarde cookies. De hecho, es una petición al explorador el guardar la secuencia `name=value` y enviarla de vuelta en cualquier otro futuro pedido del servidor. Si el explorador soporta cookies y las cookies están admitidas, cada petición de cada página subsecuente al mismo servidor va a contener la cookie. Por ejemplo, el explorador pide la página http://www.w3.org/spec.html enviando al servidor www.w3.org un pedido que se asemeja al siguiente:

```
GET /spec.html HTTP/1.1
Cookie: name=value Accept: */*
```

**explorador** → **servidor**

Este es un pedido para otra página del mismo servidor, y se diferencia del primero porque contiene la secuencia que el servidor había previamente enviado al explorador. Por donde, el servidor sabe que este pedido está relacionado con el previo. El servidor responde al enviar la página pedida, posiblemente añandiendo otras cookies también.

El valor de la cookie puede ser modificada por el servidor al enviar una nueva línea `Set-Cookie: name=newvalue` en respuesta al pedido de la página. El explorador entonces reemplaza el viejo valor con el nuevo.

La línea `Set-Cookie` no es típicamente por el servidor HTTP en sí, pero por un programa CGI. El servidor HTTP solo envía el resultado del programa (un documento precedente por el encabezado contiendo cookies) al explorador.

Las cookies también pueden ser seteadas por JavaScript o scripts similares en el explorador. En JavaScript, el objeto `document.cookie` es usado para este propósito. Por ejemplo, la instrucción `document.cookie = "temperature=20"` crea una cookie de nombre `temperature` y valor `20`.[15]

## Atributos de la *cookie*

### Caducidad

Cuando las cookies han caducado, estas no son enviadas al navegador; por lo tanto, la caducidad de las cookies puede ser pensada como un límite de tiempo en el que una de ellas puede ser usadas. La cookie puede luego ser renovada después de que este límite haya pasado. Algunos sitios prefieren que las cookies caduquen en tiempos más cortos por razones de seguridad. Las cookies no se envían al navegador si ellas están bajo estas condiciones:

- al finalizar una sesión de usuario: por ejemplo, cuando se cierra el navegador (si esta no es persistente)
- Se ha fijado una fecha de caducidad y esta ha pasado.
- La fecha de caducidad es cambiada a una fecha anterior (por el servidor)
- esta se borra por orden del usuario.

Nota: La tercer condición permite que un servidor elimine una cookie explícitamente.

### Autenticación

Muchos servidores o páginas web utilizan las cookies para reconocer usuarios que ya se hayan autenticado o para personalizar páginas web dependiendo de las opciones que un usuario seleccione. Por ejemplo, esto puede suceder cuando:

- El usuario escribe su nombre y contraseña, los cuales son enviados al servidor
- El servidor verifica la información proporcionada, y si es correcta devuelve una página de confirmación con una cookie, guardando así esta información en la computadora del usuario.
- Cuando el usuario visita una página la cual pertenece al servidor, este verifica la existencia de las cookies y luego comprueba si las cookies existentes son iguales a las que han sido guardadas en el servidor. Si hay coincidencias, el servidor puede identificar el usuario que solicitó la página.

Este es uno de los métodos de autenticación más habituales, usados por yahoo!, wikipedia, o facebook.

### Seguimiento

Otro uso de las cookies se refiere al seguimiento de una ruta (camino) que un usuario toma cuando navega a través de páginas web de un servidor o sitio. Esto también puede ser obtenido cuando se usa la dirección IP de una computadora, aunque las cookies tienen mejor precisión. Esto se puede realizar de la siguiente manera:

- Si el usuario visita una página web pero la solicitud no contiene una cookie, el servidor asume que esta es la primer visita a esa página; el servidor crea una serie de carácteres aleatorios, que luego son enviados como una cookie además de la página solicitada.
- De ahora en adelante, la cookie es enviada al servidor automáticamente por el navegador cada vez que una página se ha solicitado. El servidor envía una página como siempre, pero la fecha y hora son guardadas en un registro de la visita con la cookie.

Si luego se lee el registro, es posible identificar cuando, quién, y la secuencia en la cual un usuario accedió a que páginas.

### Cesta

Algunas páginas web, en particular páginas de compra o venta de productos permiten que usuarios guarden objetos en una "cesta virtual" así ellos estén fuera de sesión. Una lista de estos objetos puede ser almacenada en una cookie. Por ejemplo, cuando el usuario agrega un elemento a su canasta virtual, el servidor agrega el nombre de este objeto a la cookie. Sin embargo, este es un método muy inseguro y puede que la cookie sea fácilmente alterada por otro usuario. Una mejor forma podría ser que se genere una cookie de "seguimiento" aleatoria y luego usarla como una referencia en el servidor.

## Referencias


[1] Persistent client state - HTTP cookies - Preliminary specification (http://wp.netscape.com/newsref/std/cookie_spec.html)
[2] RFC 2109 y RFC 2965 - HTTP State Management Mechanism (IETF)
[3] Contrary to popular belief, cookies are good for you! (on the Internet) (http://www.theallineed.com/computers/05072901.htm)
[4] Keith C. Ivey Untangling the Web Cookies: Just a Little Data Snack (http://www.eeicommunications.com/eye/utw/98feb.html). 1998
[5] Brian Quinton. Study: Users Don't Understand, Can’t Delete Cookies (http://searchlineinfo.com/InsightExpress_cookie_study/). Direct. 18 de mayo 2005
[6] Adam Penenberg. Cookie Monsters (http://www.slate.com/id/2129656/). Slate, 7 de noviembre 2005
[7] The unofficial cookie faq (http://www.cookiecentral.com/faq/)
[8] CBS News. CIA Caught Sneaking Cookies (http://www.cbsnews.com/stories/2002/03/20/tech/main504131.shtml). March 20 2002.
[9] The Associated Press. Spy Agency Removes Illegal Tracking Files (http://www.nytimes.com/2005/12/29/national/29cookies.html). December 29 2005
[10] http://eur-lex.europa.eu/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=en&numdoc=32002L0058&model=guichett
[11] http://ec.europa.eu/information_society/policy/ecomm/doc/implementation_enforcement/annualreports/10threport/sec20041535vol1en.pdf
[12] "Can you show me what XSS cookie theft looks like?" (http://www.cgisecurity.com/articles/xss-faq.shtml#theft) (extracto de Cgisecurity Cross-Site Scripting FAQ (http://www.cgisecurity.com/articles/xss-faq.shtml))
[13] Introduction to Persistence (http://msdn.microsoft.com/library/default.asp?url=/workshop/author/persistence/overview.asp), MSDN
[14] Set the window.name property from website A then check it in website B (http://www.codingforums.com/showpost.php?p=220324&postcount=3)
[15] Cookies in JavaScript (http://www.yourhtmlsource.com/javascript/cookies.html)


## Enlaces externos

- Todo sobre las cookies (http://www.iec.csic.es/criptonomicon/cookies/cookies.html).
- La invasión de las cookies asesinas: una introducción a las cookies y sus riesgos (blog) (http://ahorapuedepegaralequipo.blogspot.com/2006/02/la-invasin-de-las-cookies-asesinas.html)

# Fuentes y contribuyentes del artículo

**Cookie** *Fuente*: http://es.wikipedia.org/windex.php?oldid=27154101 *Contribuyentes*: .Sergio, Airunp, Alhen, Balderai, Beto29, Buisqui, Chessa, Cinabrium, Comae, CommonsDelinker, Cookie, Daniel G., David0811, Davidmarco, Death Master, Diegowiki, Dodo, Farisori, Felipe Canales, Gacpro, Gaijin, Greek, Hari Seldon, Humberto, Isha, Jmieres, Jurock, Lin linao, Loco085, Lucien leGrey, MARC912374, Mae - Se pue, Mahadeva, ManuelGR, Matdrodes, Mortadelo2005, Murphy era un optimista, Máximo de Montemar, Nanuc, Netito777, Nixón, Nuvem, OceanO, Pan con queso, Poc-oban, Poco a poco, Potare, Prietoquilmes, Qwertyytrewqqwerty, Richy, Rsg, Snakefang, Tiburonsanchez, Tidsa, Tomatejc, Txuspe, Wesisnay, 184 ediciones anónimas

# Fuentes de imagen, Licencias y contribuyentes

**Archivo:Cookieeee.png** *Fuente*: http://es.wikipedia.org/windex.php?title=Archivo:Cookieeee.png *Licencia*: GNU Free Documentation License *Contribuyentes*: Jurock (usurped)

**Archivo:Firefox Cookie Manager.png** *Fuente*: http://es.wikipedia.org/windex.php?title=Archivo:Firefox_Cookie_Manager.png *Licencia*: GNU General Public License *Contribuyentes*: Jurock (usurped), Mads Ren`ai, 1 ediciones anónimas

## Licencia