

## 2.- COOKIES

El protocolo HTTP es un protocolo **sin estado**, es decir, no se guarda información acerca de las conexiones y desconexiones que se establecen entre el cliente y el servidor. Por tanto, cuando necesitamos que alguna información esté disponible entre diferentes conexiones, deberemos implementar un mecanismo que nos permita almacenar dicha información y acceder a ella. Tenemos varias soluciones posibles:

- Hacer uso de elementos de formulario ocultos.
- Almacenar la información en el servidor (sesiones).
- Almacenar la información en los equipos clientes (cookies).

Una cookie es un fichero de texto ASCII que un sitio web guarda en el equipo del usuario. Almacena información siguiendo una estructura básica de pares identificador = valor. Su tamaño es relativamente pequeño (menos de 4K) y depende del sistema operativo y del navegador el directorio en el que se almacenan.

Las cookies son reenviadas automáticamente al servidor web por el navegador cuando el usuario navega por las páginas del sitio.

La función `setcookie` permite depositar una cookie en el equipo del usuario.

```
booleano setcookie(cadena nombre [, cadena valor  
[, entero vencimiento  
[, cadena ruta  
[, cadena dominio [, booleano asegurado  
[, booleano http_únicamente]]]]])
```

Todos los parámetros excepto el nombre son opcionales.

**nombre:** Nombre de la cookie.

**valor:** Valor almacenado en la cookie.

**vencimiento:** Fecha de caducidad de la cookie (marca de tiempo Unix). Por ello, normalmente se utiliza la función `time()` (que indica el momento actual como tiempo Unix) más la duración en segundos que queremos que tenga la cookie. Por defecto dura hasta que se cierre el navegador.

**Ruta:** Ruta de acceso en el servidor donde está disponible la cookie. De forma predeterminada, igual al directorio desde donde se envió la cookie.

**dominio:** Dominio al que se reenvía la cookie. `.miSitio.com` (con un punto al principio) permite, por ejemplo, hacer que la cookie esté disponible para todos los subdominios de `miSitio.com`.

**asegurado:** Incluir `TRUE` para indicar que la cookie debe transmitirse únicamente en una conexión segura SSL (`FALSE` por defecto).

`http_unicamente`: Incluir `TRUE` para indicar que la cookie debe transmitirse únicamente por el protocolo HTTP (`FALSE` por defecto).

```
// Página 1

<?php
// La primera cookie expira al final de la sesión.
$ok1 = setcookie('nombre','Mercedes');

// Segunda cookie expira en 30 días.
$ok2 = setcookie('apellido','FLORES', time()+(30*24*3600));

// Resultado.
if ($ok1 and $ok2) {
    $mensaje = 'Cookies depositadas';
} else {
    $mensaje = 'Una de las cookies no se ha podido depositar';
}
?>

<html>
  <head><title>Página 1</title></head>
  <body>
    <div>
      <?php echo $mensaje;?><br />
      <!-- enlace hacia la página 2 -->
      <a href="pagina2.php">Página-2</a>
    </div>
  </body>
</html>
```

Las cookies se transmiten entre el navegador y el servidor web de la misma forma que las credenciales, utilizando los encabezados del protocolo HTTP. Por ello, las sentencias `setcookie` deben enviarse antes de que el navegador muestre información alguna en pantalla.

El proceso de recuperación de la información que almacena una cookie es muy simple. Cuando accedes a un sitio web, el navegador le envía de forma automática todo el contenido de las cookies que almacene relativas a ese sitio en concreto.

Desde PHP podemos acceder a esta información por medio del array `$_COOKIE`.

```
// Página 2
<html><head><title>Página 2</title></head>
<body>
  <?php
    if ( isset($_COOKIE["nombre"]) )
      echo "nombre = ".$_COOKIE['nombre'] . "<br>";
    else
      echo "nombre = <br>";
    if ( isset($_COOKIE["apellido"]) )
      echo "apellido = ".$_COOKIE['apellido'] . "<br>";
    else
      echo "apellido = <br>";
  ?>
</body>
</html>
```

## Resultado

- Presentación de la página 1:

```
Cookies depositadas  
Página-2
```

- Resultado al hacer clic en el enlace

```
nombre = Mercedes  
apellido = FLORES
```

- Resultado de un regreso, antes de treinta días, a la página 2 del mismo sitio:

```
nombre =  
apellido = FLORES
```

La cookie de duración igual a la sesión ya no existe después de abandonar la sesión y la información se pierde; la información almacenada en la otra cookie sigue estando disponible (dentro de los límites de su vida útil).

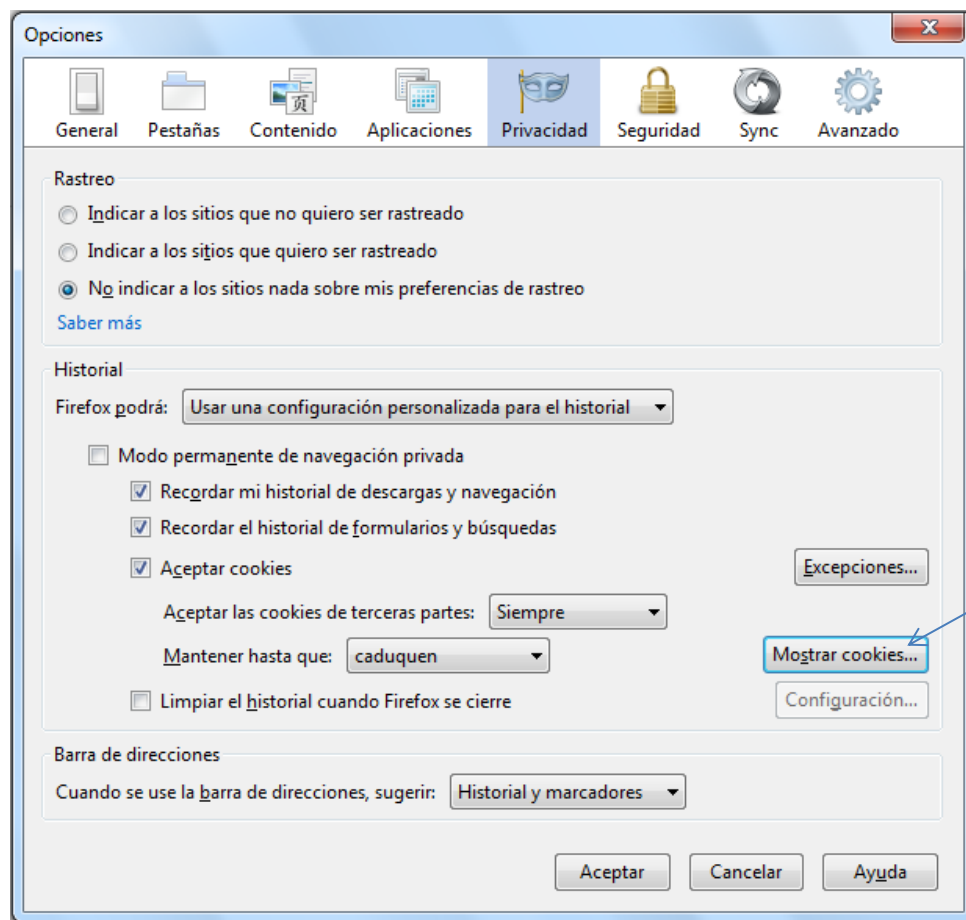
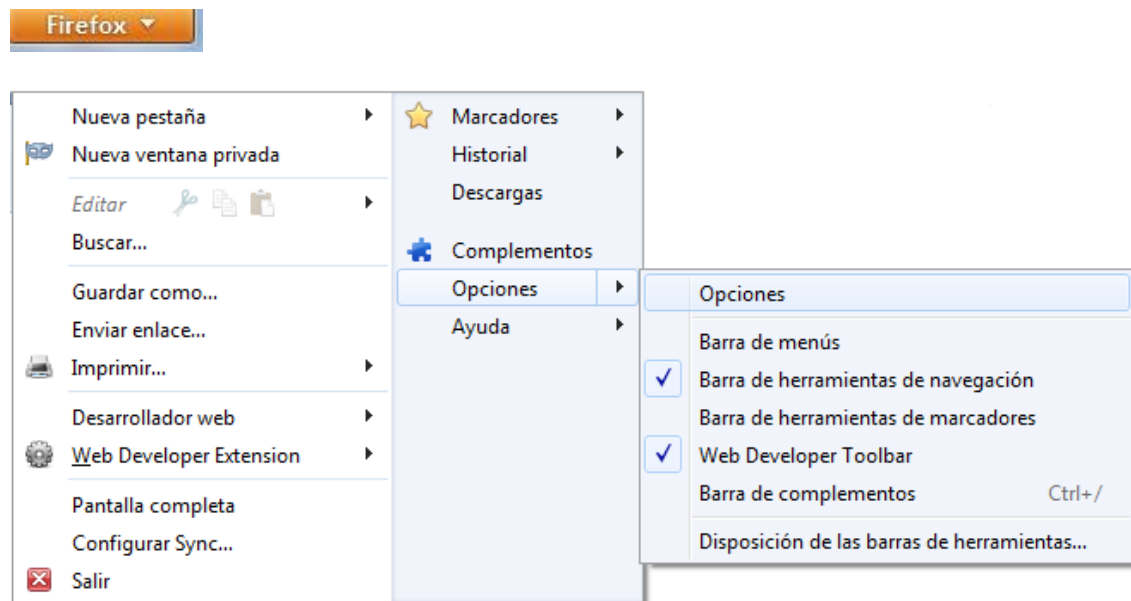
Siempre que utilices cookies en una aplicación web, debes tener en cuenta que en última instancia su disponibilidad está controlada por el cliente. Por ejemplo, algunos usuarios deshabilitan las cookies en el navegador porque piensan que la información que almacenan puede suponer un potencial problema de seguridad. O la información que almacenan puede llegar a perderse porque el usuario decide formatear el equipo o simplemente eliminarlas de su sistema.

Si una vez almacenada una cookie en el navegador quieres **eliminarla** antes de que expire, puedes utilizar la misma función `setcookie` sin pasarle valor o indicando una fecha de caducidad anterior a la actual.

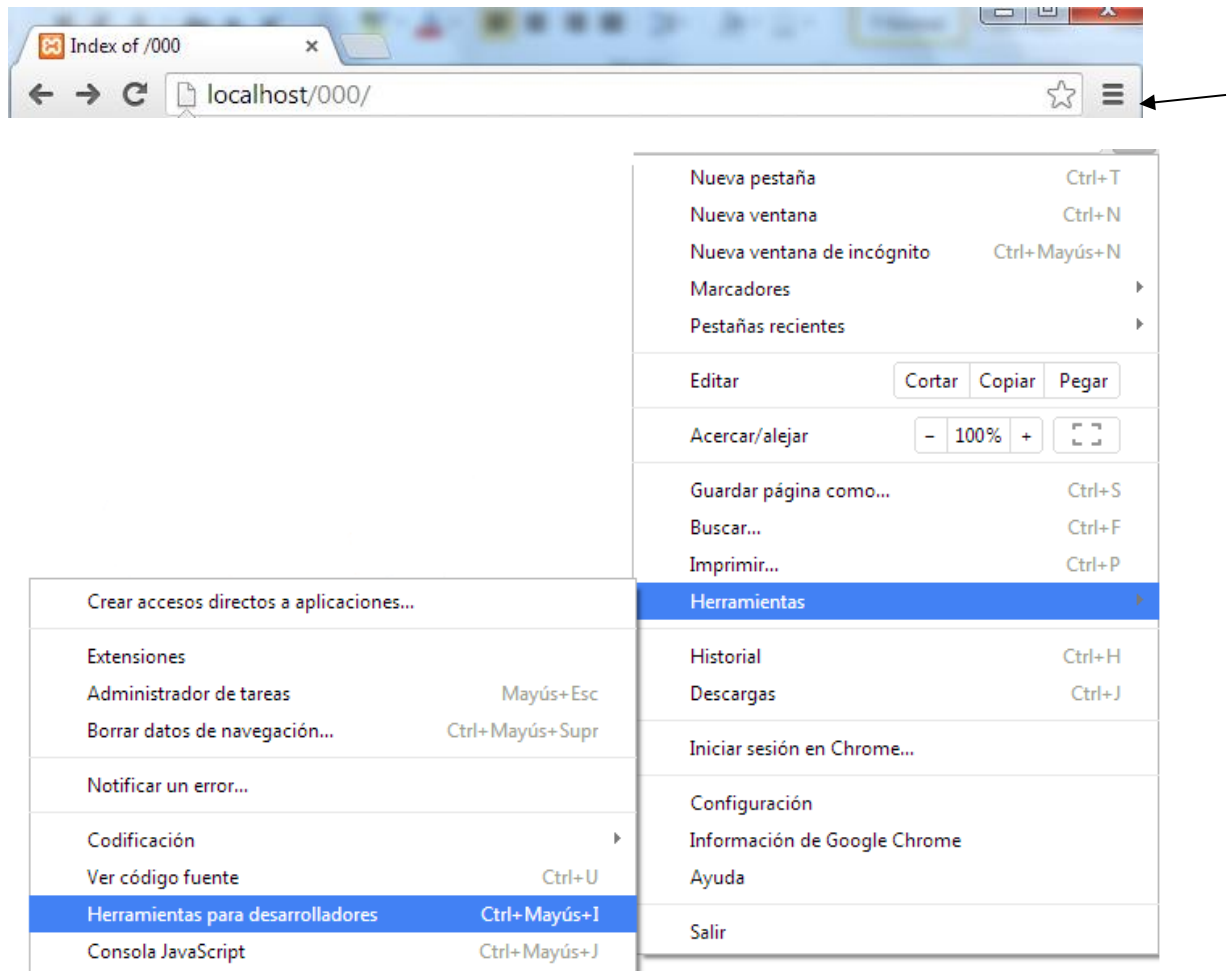
```
setcookie('nombre');
```

## Mostrar las cookies en navegador

### Firefox



## Chrome



Elements   Resources   Network   Sources   Timeline   Profiles   Audits   Console									
<div>▶ Frames</div> <div>Web SQL</div> <div>IndexedDB</div> <div>▶ Local Storage</div> <div>▶ Session Storage</div> <div>▼ Cookies</div> <div>localhost</div> <div>Application Cache</div>		<div>Name</div>	<div>Value</div>	<div>Domain</div>	<div>Path</div>	<div>Expires / Max-Age</div>	<div>Size</div>	<div>HTTP</div>	<div>Secure</div>
		apellido	FLORES	localhost	/000	Fri, 17 Jan 2014 20:47:38 GMT	14		
		nombre	Mercedes	localhost	/000	Session	14		
			</						