

INSTITUT



Unión Europea

Fondo Social Europeo

El FSE invierte en tu futuro

PHP

Cookies y Sesiones

Pepe

INSTITUT



**GENERALITAT
VALENCIANA**

Conselleria d'Educació,
Investigació, Cultura i Esport



GOBIERNO
DE ESPAÑA

MINISTERIO
DE EDUCACIÓN, CULTURA
Y DEPORTE

Continguts

1 Pròleg	3
2 Cookies	3
3 Sesiones	4
3.1 Iniciar una sesión	5
3.2 Guardar y recuperar los datos de una sesion	5
3.3 Cerrar una sesión	5
3.4 Ejemplo	5
4 Cuándo usar cookies o sesiones	6
5 Ejercicios	7

1 Pròleg

HTTP es un protocolo stateless, sin estado. Por ello, se simula el estado mediante el uso de cookies, tokens o la sesión. El estado es necesario para procesos tales como el *carrito de la compra*, *operaciones asociadas a un usuario*, etc... El mecanismo de PHP para gestionar la sesión emplea cookies de forma interna. Las **cookies** se almacenan en el navegador, y la **sesión** en el servidor web.

<http://es.php.net/manual/es/language.variables.superglobals.php>

2 Cookies

Las cookies se almacenan en el array global `$_COOKIE`. Lo que coloquemos dentro del array, **se guardará en el cliente**. Hay que tener presente que el cliente puede no querer almacenarlas.

Existe una limitació de 20 cookies por dominio y 300 en total en el navegador.

En PHP, para crear una cookie se utiliza la función `setcookie`:

```
<?php
setcookie(nombre [, valor [,expira [,ruta [,dominio [,seguro [,httponly
]]]]]);
setcookie(nombre [, valor = "" [, opciones = [] ]] )
?>
```

Destacar que el nombre no puede contener espacios ni el caracter “;”. Respecto al contenido de la cookie, no puede superar los 4 KB. Por ejemplo, mediante cookies podemos comprobar la cantidad de visitas diferentes que realiza un usuario:

```
<?php

$num_accesos = 0;
if(isset($_COOKIE['acceso'])){
    $accesos=$_COOKIE['acceso']; //Recuperamos la cookie. Se ha creado
    previamente.
    setcookie('acceso',num_accesos++); //Incrementamos y guardamos.
}
?>
```



Si queremos ver que contienen las cookies que tenemos almacenadas en el navegador, se puede comprobar su valor en *Dev Tools* → *Application* → *Storage*

El tiempo de vida de las cookies puede ser tan largo como el sitio web en el que residen. Ellas seguirán

ahí, incluso si el navegador está cerrado o abierto.

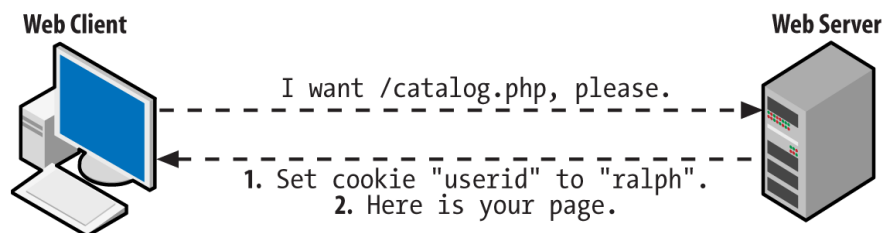
Para borrar una cookie se puede poner que expiren en el pasado:

```
<?php
setcookie(nombre, "", 1) // pasado
?>
```

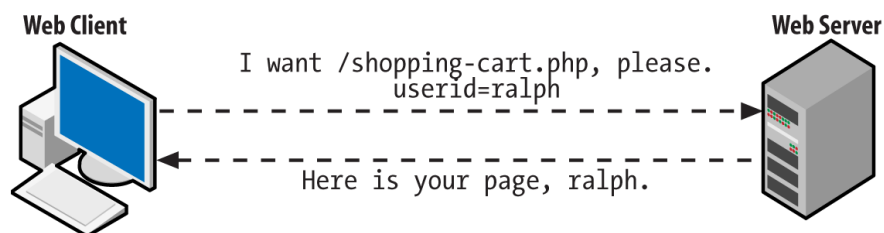
O que caduquen dentro de un periodo de tiempo determinado:

```
<?php
setcookie(nombre, valor, time() + 3600) // Caducan dentro de una hora
?>
```

First Request



Second Request



(a) cookies

Se utilizan para:

- Recordar los inicios de sesión.
- Almacenar valores temporales de usuario.
- Si un usuario está navegando por una lista paginada de artículos, ordenados de cierta manera, podemos almacenar el ajuste de la clasificación.

3 Sesiones

`$_SESSION` es un array especial utilizado para guardar información a través de los *requests* que un usuario hace durante su visita a un sitio web o aplicación. La información guardada en una sesión

puede llamarse en cualquier momento mientras la sesión esté abierta.

Las operaciones que podemos realizar con la sesión son:

3.1 Iniciar una sesión

- Para manejar datos en una sesión, primero deberemos crearla con la función `session_start()`.

3.2 Guardar y recuperar los datos de una sesión

- utilizaremos el array `$_SESSION` para guardar y recuperar datos de la sesión. Por ejemplo, si queremos guardar el login del usuario que nos ha enviado por un formulario, pondríamos algo como:

```
$_SESSION["loginUsuario"] = $_REQUEST["login"];
```

3.3 Cerrar una sesión

- usamos la función `session_destroy()`. Es aconsejable *borrar* antes a *mano* todas las variables que hayamos creado en la sesión, para que no se queden ocupando memoria. Si queremos eliminar algún dato de la sesión, usaremos la función `unset` con dicho dato del array:

```
unset($_SESSION["loginUsuario"]);
```

3.4 Ejemplo

Veamos un ejemplo un poco más completo de todo esto. Imaginemos que queremos guardarnos en sesión el e-mail de un usuario que nos llega a través de un formulario. En la página donde recogemos los datos de ese formulario haríamos algo así:

```
// Esto nos permitirá acceder al array $_SESSION
session_start();

// Comprobamos si llega un e-mail en la petición
if (isset($_REQUEST['email']))
{
    // Almacenamos el e-mail en una casilla del array $_SESSION,
    // con el nombre que queramos (por ejemplo, 'emailUsuario')
    $_SESSION['emailUsuario'] = $_REQUEST['email'];
}
```

```
else {  
    // Si no existe, podemos redirigir a otra página, por ejemplo  
    header("Location:inicio.php");  
}  
// ... Resto del código de la página
```

Finalmente, cuando queramos eliminar estos datos de sesión del usuario, podemos llamar a `session_destroy`. Es conveniente también eliminar una a una las variables, o limpiar el array de sesión para liberar la memoria:

```
session_start();  
if (isset($_SESSION['emailUsuario']))  
    unset($_SESSION['emailUsuario']);  
// También serviría esto:  
// $_SESSION = array();  
session_destroy();
```

4 Cuándo usar cookies o sesiones

Ya hemos visto estos dos mecanismos de compartir y actualizar información entre cliente y servidor, pero... ¿cuándo es conveniente utilizar cookies y cuándo sesiones? Básicamente debemos regirnos por estas sencillas reglas:

- Usaremos **cookies** para almacenar **información simple** (texto) y no relevante para el funcionamiento de la web (porque los navegadores pueden desactivar las cookies). Por ejemplo, las últimas búsquedas que ha hecho un usuario en una web, o el login del usuario para recordárselo en el formulario la próxima vez que intente loguearse (**aunque el login también deberemos almacenarlo en sesiones para recordar quién ha entrado hasta que cierre sesión**).
- Usaremos **sesiones** cuando queramos almacenar **datos complejos**, o **cruciales** para el funcionamiento de la web, ya que no pueden ser desactivadas por el navegador. Por ejemplo, el login del usuario (no el password) es útil almacenarlo en sesiones, para tener la certeza de poder identificarlo en cada página que visita (si estuviera en cookies, podrían desactivarse y perder quién ha entrado). Los elementos de un carro de la compra que tengamos añadidos de una tienda virtual, también se deberían almacenar en sesiones, pues son (o pueden ser) datos complejos, no simple texto.
- **No usaremos ni una ni otra para almacenar información privada del usuario, como por ejemplo contraseña, número de tarjeta de crédito, etc.**

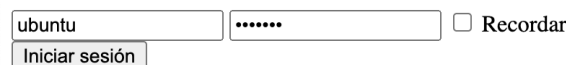
5 Ejercicios

5.0.1 ejcookies.php

Realizar una aplicación que compruebe si existe la *cookie* “user” tiene datos, vuestro nombre, en caso de estar vacía que la cree con una caducidad de *1000*. En la siguiente ejecución debe de aparecer vuestro nombre en el navegador.

5.0.2 recordar.php

Realizar un formulario de entrada para los datos de *login* y *pass* con un *check recordar*. Si está seleccionado el *check* a la siguiente sesión debe cargar los datos de manera automática.



ubuntu ☐ Recordar
Iniciar sesión

(b) recordar.php

5.0.3 carro.php

Crea una carpeta llamada *carro* con una página llamada *carro.php* con una lista de enlaces de diferentes artículos y su precio. Por ejemplo:

```
Zapatillas Nike (60 euros)
Sudadera Domyos (15 euros)
Pala de pádel Vairo (50 euros)
Pelota de baloncesto Molten (20 euros)
```

Puedes alimentar la lista desde un array de artículos previamente definido en PHP, como este:

```
$articulos = array(
    array("id" => 1, "nombre" => "Zapatillas Nike", "precio" => 60),
    array("id" => 2, "nombre" => "Sudadera Domyos", "precio" => 15),
    array("id" => 3, "nombre" => "Pala de pádel Vairo", "precio" => 50),
    array("id" => 4, "nombre" => "Pelota de baloncesto Molten", "precio"
        => 20)
);
```

Cada vez que el usuario haga clic en un artículo, se enviará a la propia página, recogerá el código o id del artículo (que se le enviará como parámetro) y con ello buscará el artículo en el catálogo y acumulará en sesión el total de lo que ha ido comprando, junto con un listado de los artículos que ha seleccionado, para mostrarlo todo por pantalla. Por ejemplo:

Lista de Artículos

- [Zapatillas Nike](#) (60 euros)
- [Sudadera Domyos](#) (15 euros)
- [Pala de pádel Vairo](#) (50 euros)
- [Pelota de baloncesto Molten](#) (20 euros)

Carro de Compras

- Zapatillas Nike - 60 euros
- Sudadera Domyos - 15 euros
- Pala de pádel Vairo - 50 euros
- Pelota de baloncesto Molten - 20 euros
- Pala de pádel Vairo - 50 euros
- Pala de pádel Vairo - 50 euros
- Sudadera Domyos - 15 euros
- Zapatillas Nike - 60 euros

Total: 320 euros

(c) carro.php

Añade un enlace a la página que sea “Vaciar carro” que limpie el carro y deje así el total a 0.

5.0.4 calificaciones.php

Crear una pequeña aplicación que permita la gestión académica del módulo de DWES. Interesa almacenar las notas de cada trimestre y mostrar un informe con las notas la media y el nombre de los alumnos. También debe haber un botón/enlace para borrar los datos.

Calificación Alumnos

<input type="text" value="Nombre alumno"/>	<input type="button" value="Noí"/>	<input type="button" value="Noí"/>	<input type="button" value="Noí"/>	<input type="button" value="Añadir"/>
--	------------------------------------	------------------------------------	------------------------------------	---------------------------------------

Lista de Alumnos

Nombre	Primer	Segundo	Segundo	Media
Pepe	6	4	5	5
Armando Bronca	6	5	6	6
Elver Galarga	5	4	3	4
Rosa Melano	3	3	6	4

[Borrar notas...](#)

(d) calificaciones.php