PROGRAMACIÓN WEB EN ENTORNO SERVIDOR UD-12

PERSISTENCIA EN PHP

ÍNDICE

[VARIABLES DE SERVIDOR 3](#_Toc147867406)

[A. $\_SERVER[] 3](#_Toc147867407)

[VALIDACIÓN DE DATOS 4](#_Toc147867408)

[TEXTO IMPRIMIBLE 4](#_Toc147867409)

[INTRODUCCIÓN DE <br /> 4](#_Toc147867410)

[FORMATEO SALIDA CON PRINTF 5](#_Toc147867411)

[FORMATEO SALIDA CON NUMBER\_FORMAT 5](#_Toc147867412)

[FUNCIONES Y CONSTANTES MATEMÁTICAS 6](#_Toc147867413)

# INTRODUCCIÓN

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.

No confundir las session con las cookies. Son muy similares y es fácil confundirlas. Las cookies son unas variables que también se guardan en el navegador del cliente, pero si podrá verse su contenido y además su vida será más larga.

# 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 limitación de 20 cookies por dominio y 300 en total en el navegador.

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

Texto

Descripción generada automáticamente

Sólo es obligatorio el primer campo, el del nombre, el resto de campos son opcionales.

Por ejemplo:



El nombre no puede contener espacios ni el carácter ;. El contenido de la cookie, no puede superar los 4 KB.

Para ver las cookies de un sitio web, en Chrome nos metemos en **Herramientas para desarrolladores** y seleccionamos **Aplicación** y después **Cookies.**

Interfaz de usuario gráfica

Descripción generada automáticamente

El funcionamiento de las cookies es muy sencillo:

1. Se realiza una petición web a un servidor a través de un cliente web.
2. El servidor setea una (o varias) cookie para ese cliente web y devuelve la página.
3. En siguientes peticiones desde ese mismo cliente web no sólo se mandará la web a la que se quiere acceder sino también la cookie (o cookies) que se ha establecido.

Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente

Lo bueno de las Cookies es que, aunque el usuario abandone nuestro sitio y vuelva más adelante podremos recuperar la información. A no ser que haya borrado las Cookies de nuestra web.

A pesar de la mala fama que tienen son útiles y seguras para los programadores, otra cosa es que exista empresas sin escrúpulos que quieran aprovecharse de sus virtudes. Poseen muchos usos:

* Guardar tu configuración para mejorar la experiencia (autologin, cambiar idioma que elegiste la primera vez, la última búsqueda que realizaste, evitar mostrar avisos que ya aceptaste…).
* Usos comerciales (sugerencias personalizadas dependiendo de que buscaste en la última ocasión, guardar tu carrito de compra…)
* Crear estadísticas.

Veamos unos ejemplos:

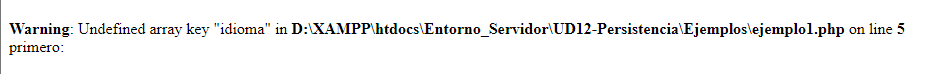


Este código va a crear una cookie con el nombre idioma y el valor ‘es’, además tendrá un tiempo de caducidad de 60 segundos.

Para imprimir la cookie necesitamos una llamada al array $\_COOKIE:



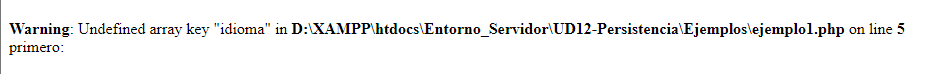
La primera vez que accedemos al sitio web esto nos provocará un error porque la cookie se crea en tiempo de respuesta desde el servidor y no envío desde el cliente, dicho de otra manera, el cliente no ha mandado ninguna cookie por lo que estamos intentando acceder a un campo que no está definido:



La siguiente vez que carguemos la página sí que se nos imprimirá el mensaje de forma correcta:



Tras 60 segundos la cookie caducará y si entramos en el sitio volveremos a recibir el mensaje de error anterior:



Ahora vamos a cambiar el idioma:



Pero la primera vez que recargamos la página tras esto, recibimos este mensaje:



De nuevo, esto se debe a que la cookie cuando se ha hecho el envío de información desde el cliente seguiría teniendo el valor ‘es’ almacenado. Si realizamos un refresco de la página sí que obtendremos el valor ‘pt’:

Texto

Descripción generada automáticamente

Vamos a eliminar la cookie, para ello vamos a usar unset($\_COOKIE[‘nombreCookie’]):



Si ahora recargamos la web recibimos de nuevo el mensaje de error porque hemos eliminado la cookie:



# SESIONES

La sesión añade la gestión del estado a HTTP, almacenando en este caso la información en el servidor.

Cada visitante tiene un ID de sesión único yse almacena defecto en una cookie denominada PHPSESSID. Si el cliente no tiene las cookies activas, el ID se propaga en cada URL dentro del mismo dominio.

Cada sesión tiene asociado un grupo de datos mediante el array global $\_SESSION, en el que podemos almacenar y recuperar información.

Texto

Descripción generada automáticamente con confianza media

Vamos a crear una sesión para ello usamos el método session\_start() y a imprimir por pantalla el ID que se ha generado con el método session\_id():



Si ahora accedemos al sitio web obtendremos este mensaje:



En este caso recibimos respuesta desde el primer momento y esto se debe a que la sesión se crea en el servidor por lo que en la respuesta que nos da el servidor por primera vez ya se sabe la sesión que vamos a tener.

Podemos crear atributos en el array $\_SESSION de la siguiente manera:





Si queremos eliminar un atributo debemos usar unset($\_SESSION[‘nombreAtributo’]).

Para cerrar una sesión utilizamos el método session\_destroy().

