

Tabla de contenido

13/01/2020 (Presencial 4): Unidad 3: Desarrollo de aplicaciones web con PHP.	2
AUTENTICACIÓN DE USUARIOS Y CONTROL DE ACCESOS.....	2
MECANISMOS DE AUTENTIFICACIÓN	2
COOKIES	3
HERRAMIENTAS DE DEPURACIÓN DE CÓDIGO.....	3
EJERCICIOS PROPUESTOS DE COOKIES	4
19_COOKIES_00_COLOR_FONDO	4
19_COOKIES_01	4
19_COOKIES_02	5
19_COOKIES_03_LISTA_TAREAS	6
EJERCICIOS PROPUESTOS DE SESIONES.....	6
20_SESIONES_00_00.....	6
20_SESIONES_01	6
20_SESIONES_02: problema: acceso directo a un script sin login ..	6
20_SESIONES_03 (solución al problema anterior)	7

13/01/2020 (Presencial 4): Unidad 3: Desarrollo de aplicaciones web con PHP.

En esta tutoría hemos repasado la unidad 3.

Las cookies y sesiones son elementos que se utilizan mucho en el desarrollo de aplicaciones web, por lo que es importante conocerlas y saber cómo programarlas.

AUTENTICACIÓN DE USUARIOS Y CONTROL DE ACCESOS

MECANISMOS DE AUTENTIFICACIÓN

En la teoría de la plataforma se indica la forma de establecer mecanismos de autenticación con Linux. A continuación os explico cómo hacerlo en Windows.

En Windows:

desde la línea de comandos:

En el directorio xamp\apache\bin está el ejecutable htpasswd.exe

```
htpasswd -c apache.usr alumno
```

Esta instrucción crea el archivo usr añadiendo el usuario alumno. Pide teclear la contraseña dos veces.

Con la instrucción `type apache.usr` se comprueba el contenido del archivo. La contraseña aparece encriptada con el algoritmo MD5.

También con `notepad usr`

Si necesitas agregar más de un usuario ejecuta la instrucción:

```
htpasswd -b apache.usr invitado  
htpasswd -b apache.usr dwes abc123.
```

Después de la opción -b va el nombre del archivo de usuarios previamente creado seguido del usuario y por último la contraseña, en el caso del ejemplo el usuario y contraseña son iguales.

Una vez creado el archivo de usuarios, creamos el fichero .htaccess en el directorio donde va a estar nuestra aplicación con las directivas específicas:

```
#En la directiva AuthName puedes poner lo que quieras  
AuthName "Contenido restringido"  
#Indicamos que el tipo de autenticación es básico  
AuthType Basic
```

```
#hay que indicar la ruta absoluta al archivo apache.user
AuthUserFile C:\xampp\bin\apache.user
#indica que los únicos usuarios que pueden entrar son
alumno e invitado
require user alumno invitado dwes
```

Después hay que asegurarse de que en el archivo `http.conf` (del directorio `conf`) la directiva `AllowOverride` tiene el valor `"ALL"` para que los ficheros `.htaccess` no sean ignorados. (buscar **"AllowOverride controls what directives may be placed in .htaccess files."**, está ahí, por defecto está con valor `all`)

COOKIES

Disponemos de varias formas para ver las cookies que tenemos almacenadas en nuestro navegador.

CHROME:

- ir a Configuración => Configuración avanzada => Privacidad y seguridad => Configuración del sitio web => Cookies y datos de sitios => Ver todas las cookies y datos de sitios web. Buscar localhost, y pulsando en la flecha que aparece a la derecha veremos todas las cookies guardadas de localhost.
- Otra forma de ver las cookies guardadas, es activando las herramientas de desarrollador (F12) y seleccionar la pestaña "Application". En el panel de la izquierda con título "Storage", pulsar sobre "Cookies" y ver que tiene localhost.

FIREFOX:

- En las nuevas versiones de Firefox, desde la configuración, no se pueden ver en detalle las cookies guardadas, pero, desde nuestra página abierta en el navegador, podemos activar la consola de Desarrollador CTRL + SHIFT + K, seleccionar la pestaña Almacenamiento, y desplegar las cookies.

HERRAMIENTAS DE DEPURACIÓN DE CÓDIGO

Es muy importante poder depurar el código.

Os dejo algunos de enlaces para aprender a configurar y trabajar con Xdebug

<http://wiki.netbeans.org/HowToConfigureXDebug>

<https://netbeans.org/kb/docs/php/debugging.html#options>

<http://microarticulos.com/instalar-depurador-php-xdebug-para-netbeans-en-windows-usando-xampp/>

<https://www.youtube.com/watch?v=XYcwmvBCqUM>

EJERCICIOS PROPUESTOS DE COOKIES

Se adjunta la solución de los ejercicios propuestos.

Una cookie consta de un nombre, un valor, una fecha de expiración y un servidor. Una cookie está limitada a 4KB.

[19_COOKIES_00_COLOR_FONDO](#)

Este ejercicio consta de dos páginas:

Index.php

En la primera tenemos un formulario en el que los usuarios van a decidir el color de fondo que va a tener la página. El color elegido lo guardamos en una cookie. El color por defecto será blanco.

En esta página se comprueba si existe la cookie, y si es así pone su valor como color de fondo.

Pagina2.php

En esta página se crea la cookie propiamente dicha usando la función **setcookie()**. La llamada a la función setcookie **debe hacerse antes de cualquier etiqueta HTML** o no funcionará.

Ponemos como nombre de la cookie la palabra "color" y como valor el valor que se haya asignado en el formulario.

El valor de expiración lo calculamos utilizando la función time() que nos devuelve la fecha actual en segundos y le sumamos el producto de 60 segundos * 60 minutos * 24 horas * 365, es decir un año.

Le indicamos como directorio la sintaxis "/" lo que indica que la cookie se crea a nivel de sitio.

Observa que cada vez que arrancamos la página el color de fondo que aparece es el último que se seleccionó

[19_COOKIES_01](#)

Crear una página en la que se comprueba si existe una cookie con el nombre "usuario".

Si no existe, se crea dinámicamente un formulario en el que se pedirá que se introduzca un nombre.

Al pulsar el botón del formulario, se va a otra página donde se crea la cookie con una duración de un año, mostrando un mensaje indicando que la cookie ha sido creada y un enlace para volver a la página principal.

Si existe la cookie, el formulario no se ve y en su lugar se muestra un mensaje con el nombre de usuario que contiene la cookie y un enlace a una tercera página en la que se borrará la cookie.

En la tercera página se borrará la cookie, mostrando un mensaje indicándolo y se mostrará un enlace a la página principal.

Este ejercicio consta de tres archivos:

- **Index.php:** En este script se comprueba si existe la cookie "usuario". Si es así, muestra un mensaje de saludo y un enlace a pagina3.php. Si no existe se crea de forma dinámica un formulario en el que se al usuario que teclee un nombre (tiene el atributo required), y también tiene un botón para enviar los datos del formulario. El action tiene como valor pagina2.php
- **Pagina2.php:** Guarda la cookie usuario con el valor que le ha llegado del formulario con una duración de un año. En el código HTML, se muestra un mensaje indicando que se ha creado la cookie y un enlace para ir a la página principal.
- **Pagina3.php:** Borra la cookie poniendo una fecha anterior a la actual. Muestra un mensaje indicando que se ha borrado y un enlace a la página principal.

[19_COOKIES_02](#)

Ejemplo: preferencias de usuario

- El usuario debe seleccionar en un formulario sus preferencias en cuanto a color de fondo, color de letra, y tipografía (entre tres opciones). Además debe introducir el nombre y los apellidos.
- Al enviar el formulario se redirige a una página en la que se le saluda por sus nombre y que está configurada con los colores y tipo de letra elegido. Las opciones seleccionadas se guardarán en cookies, de forma que si se vuelve a ejecutar el script principal, si se detecta que hay cookies, se redirige automáticamente a esta segunda página, configurada con las selecciones realizadas.
- La segunda página contiene un enlace que al pulsarlo redirige a la primera página, borrando además las cookies con las opciones elegidas.

[19_COOKIES_03_LISTA_TAREAS](#)

- Hacer una aplicación web de una sola página en la que figure un formulario con un campo de tipo input en el que se debe teclear una tarea a realizar.
- Al pulsar el botón Añadir, se van mostrando en la parte inferior de la página en forma de lista, todas las tareas que se han ido añadiendo, y al lado de cada una de ellas, un enlace que lleve a la misma página con el texto Borrar.
- Si se pulsa el enlace Borrar se elimina la tarea que correspondía a ese enlace.
- La página debe conservar las tareas usando COOKIES.

[EJERCICIOS PROPUESTOS DE SESIONES](#)

Se adjunta la solución de los ejercicios propuestos.

[20_SESIONES_00_00](#)

En el siguiente ejemplo se muestra por pantalla el identificador de sesión que se nos ha asignado, el nombre de la sesión (que también es el nombre de la cookie que se emplea para guardar el identificador de la sesión, y por último se muestran los parámetros de la cookie que almacena el identificador de la sesión.

[20_SESIONES_01](#)

Escribir una pequeña aplicación web en PHP de dos páginas.

- La idea es que un usuario introduzca un nombre y contraseña válidos y seleccione o no una opción "Recordar datos".
- Al pulsar el botón enviar, irá a otra página en la que se mostrará el nombre introducido, y un mensaje indicando si se recordarán los datos introducidos o no al volver a la primera página (en función de lo que hubiera seleccionado).
- Al pulsar el enlace de la segunda página, vuelve a la primera en la que se mostrará el formulario vacío si no seleccionó la opción "Recordar datos", o con los datos que hubiera rellenado al principio en caso de haberla marcado.
- Los datos se guardarán en una sesión.

[20_SESIONES_02: problema: acceso directo a un script sin login](#)

Trabajando con sesiones se nos puede plantear el siguiente problema:

- Imaginemos que tenemos una aplicación web con tres páginas. En la primera hay un formulario de login en el que simplemente se pide al

usuario que introduzca su nombre, para en la segunda página mostrar los datos de la sesión, y en la tercera borrarlos.

- ¿Qué crees que ocurrirá si una vez logueado un usuario intenta acceder a la página 2 o 3 directamente (le puede salir en el historial), sin pasar por el login?

La aplicación no debería dejarle acceder a esa página directamente, pero tenemos datos de sesión guardados que indican que se ha logueado en algún momento y por tanto puede acceder.

[20_SESIONES_03 \(solución al problema anterior\)](#)

- Una posible solución muy sencilla de implementar es crear una variable de sesión o una cookie que se encargue de controlar si ha pasado por el proceso de login o no.
- En las páginas 2 y 3 se comprueba que esa variable existe, y de ser así se borra para evitar que pueda acceder directamente a la página.