RESUMEN DE PHP XI – Sesiones y Cookies

HTTP es un protocolo sin estado. Esto quiere decir que el protocolo no dispone de un método para conservar el estado entre dos peticiones de un mismo usuario. PHP dispone de una forma para implementar las sesiones generando un identificador de sesión aleatorio que se almacena en el lado del cliente en forma de cookie, o si el cliente tiene deshabilitada las cookies se pasa al servidor por la URL. Este identificador actúa como una clave y nos permite almacenar variables de sesión que se guardan en el servidor.

Cookies

Una cookie es un fragmento de información que queda almacenada en el ordenador del usuario. Las cookies contienen pares de clave/valor y solo son accesibles por el dominio que las crea. En PHP hay dos formas de crear cookies en el ordenador del usuario, mediante la función **setcookie()** o enviando una cabecera en la respuesta HTTP. Para poder utilizar los dos métodos no se deben haber enviado datos al flujo de salida. Vamos a ver como crear cookies utilizando los dos métodos y como recuperar la información pasada por cookies desde el servidor con PHP.

💻 Cabecera Set-Cookie:

El protocolo HTTP cuenta con una cabecera para establecer cookies y podemos enviarla con la función **header()**. La cabecera en cuestión tiene la siguiente sintaxis:

Set-Cookie: NOMBRE=VALOR; expires=FECHA; path=RUTA; domain=NOMBRE_DOMINIO; secure

De esta forma se crea una cookie con el nombre NOMBRE y con el valor VALOR. Los demás parámetros son opcionales, el campo expires establece el tiempo que se mantendrá la cookie en el lado del cliente, path y domain los podemos utilizar para restringir el acceso a la cookie a ciertos dominios o rutas, y secure establece si la cookie se envia a través de una conexión segura HTTPS.

```php
<?php
        header('Set-Cookie: nombre=Francisco;');
?>
```

En este ejemplo hemos creado una cookie en el ordenador del usuario con la clave *nombre* y el valor *francisco*.

💻 setcookie():

Esta función recibe como parámetros los mismos que la cabecera Set-Cookie, un nombre, un valor, una fecha, una ruta, un dominio y un booleano para establecer cookies seguras.

setcookie() define una cookie para ser enviada junto con el resto de las cabeceras HTTP. Como otras cabeceras, las cookies deben ser enviadas antes de cualquier salida desde su script (esta es una restricción de protocolo). Esto requiere que coloques las

llamadas a esta función antes de cualquier salida, incluyendo las etiquetas *<html>* y *<head>* así como cualquier espacio en blanco. Si existe salida antes de llamar esta función, setcookie() fallará y devolverá FALSE. Si setcookie() se ejecuta con éxito, devolverá TRUE. Esto no indica si el usuario aceptó la cookie.

```php
<?php
        setcookie('nombre','Francisco');
?>
```

El mismo ejemplo utilizando la función.

Acceder a las cookies desde PHP:

Podemos acceder a las cookies definidas mediante la variable super-global $_COOKIE, un array asociativo que almacena los valores mediante las claves definidas en las cookies. De esta forma en nuestro ejemplo podemos acceder a la cookie creada de la siguiente forma.

```php
<?php
        $nombre=$_COOKIE['nombre'];
        //Imprimirá Francisco
        echo $nombre;
?>
```

Podemos eliminar las cookies si establecemos la misma cookie con una fecha de caducidad pasada. El problema de utilizar cookies para guardar información referente al usuario es que los datos se almacenan en el ordenador del usuario, y pueden no estar disponibles si el usuario ha deshabilitado el uso de cookies en su navegador. Esta es la principal razón por las que se utilizan las sesiones para guardar datos entre distintas peticiones del mismo usuario.

Sesiones

PHP utiliza por defecto cookies con sesiones. Esto quiere decir que si el navegador del usuario acepta cookies se definirá una para almacenar el identificador de sesión. Si el navegador del usuario no acepta cookies, el identificador se pasará como un parámetro más por la URL. Si queremos utilizar este último método siempre, debemos activar la directiva **session.use_trans_sid** en el archivo de configuración de PHP, aunque no está muy recomendado.

Los pasos que hay que seguir para utilizar sesiones son los siguientes:

1. Iniciar una sesión.

2. Registrar variables de sesión.

3. Utilizar estas variables.

4. Anular las variables registradas y eliminar la sesión.

Iniciando una sesión:

Para iniciar una sesión basta con llamar a la función **session_start()** antes de enviar cualquier carácter al flujo de salida siempre que queramos utilizar sesiones. Esta función comprueba si hay un identificador para el usuario, y si lo hay carga en la variable super-global **$_SESSION** todas las variables de sesión registradas.

```
<?php
        session_start();
?>
```

Registrando variables de sesión:

Para registrar variables de sesión basta con definir un elemento en el array asociativo $_SESSION de la siguiente forma: **$_SESSION['clave']=valor;**

```
<?php
        session_start();
        $_SESSION['nombre']='Francisco';
?>
```

Utilizando variables de sesión:

Antes de poder utilizar las variables de sesión definidas tenemos que iniciar la sesión. Una vez iniciadas ya podemos acceder a las variables desde el array asociativo **$_SESSION['clave']**. Si utilizamos objetos como variables de sesión debe incluir antes la definición de clase.

Ej: nos hemos creado una clase coche y hemos guardado en una variable de sesión un objeto de dicha clase, con lo cual si queremos recuperar dicho objeto y usarla tendremos que incluir dicha clase (include())

```
<?php
        session_start();
        //Imprimirá Francisco
        echo $_SESSION['nombre'];
?>
```

Anular las variables de sesión y eliminar la sesión:

Para eliminar una variable de sesión cuando ya no necesitemos utilizarla, podemos llamar a la función **unset()** pasandole como argumento la variable que queramos eliminar. No intentéis borrar el array **$_SESSION** completo porque puede deshabilitar las sesiones. Si queremos anular todas las variables podemos crear un nuevo array vacío.

Para destruir la sesión llamamos a la función **session_destroy()**.

```
<?php
        session_start();

        //Eliminamos un par clave/valor
        unset($_SESSION['nombre']);
```

```
        //Eliminamos todas las variables de sesión
        $_SESSION = array();

        //Destruimos la sesión
        session_destroy();
?>
```

Configurar el control de sesiones:

En el archivo de configuración de PHP existe un conjunto de directivas para configurar las sesiones. Ya hemos visto la directiva *session.use_trans_sid* que permitia establecer la forma de pasar el id a través de las URLs.

Algunas directivas que podemos configurar son:

- session.auto_start: Inicia las sesiones automáticamente, con el inconveniente que no se pueden almacenar objetos como variables de sesión.
- session.cache_expire: Establece la duración en minutos de la sesión.
- session.cookie_lifetime: Indica el tiempo que permanecerá la cookie de sesión en el navegador del usuario. Por defecto está 0, que indica que se borrará la cookie cuando el navegador se cierre.
- session.use_cookies: Establece las sesiones para que utilicen cookies. Por defecto viene activado.

En definitiva, utilizando las sesiones en PHP podemos implementar la autentificación de usuarios, por ejemplo, para mostrar ciertos contenidos dependiendo del usuario que esté utilizando la aplicación. Bastaría con definir una variable de sesión con el nombre o el id del usuario registrado en el sistema.