

41311 Laboratorio de Interfaces de Redes

3º Ingeniero Técnico en Informática de Gestión

Práctica 5 - PHP avanzado: Sesiones

1. Objetivos

En esta práctica veremos cómo emplear la gestión de sesiones en PHP para conservar el valor de variables de unos scripts a otros.

El API sobre sesiones en PHP viene en:

http://www.tlm.unavarra.es/~daniel/docencia/lir/lir03_04/manuales/php/ref.session.html

2. Historia mediante variables de sesión

El empleo de sesiones nos permite conservar variables de unos scripts PHP a otros o de una ejecución a otra del mismo script para el mismo usuario.

En esta sección se plantea crear un script que vaya almacenando en una variable de la sesión los URLs del servidor que ha ido solicitando el usuario. Para ello, en cada página se llama al principio a ese script (o a una función del mismo) el cual añade el URL del documento actual a un array con los URLs visitados, array que se guarda como variable de sesión. Para ver el histórico de los URLs visitados podemos crear otra función en PHP a la que se le llama al final del documento para sacar un pie para el mismo con la lista de URLs en el orden en que se han ido visitando.

Checkpoint 1

Crean varias páginas que empleen esas dos funciones y muestren al profesor de prácticas que les funciona

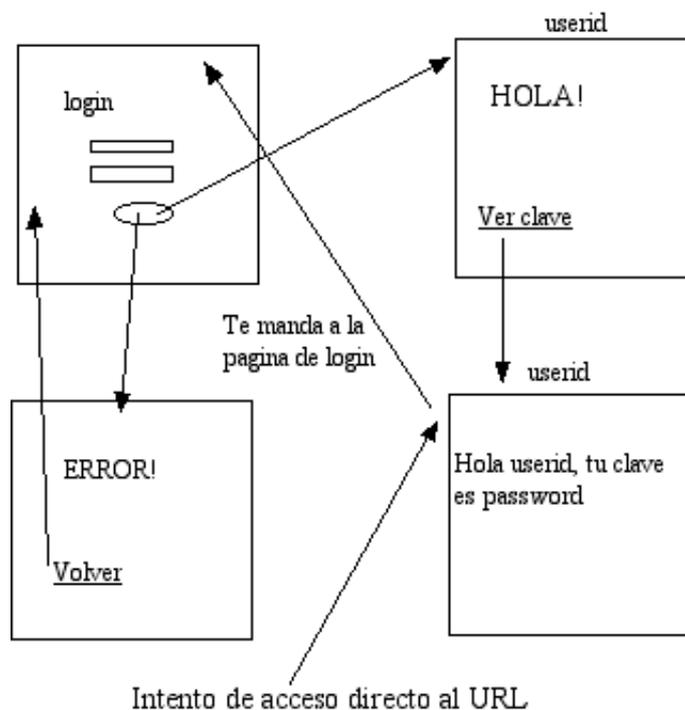
3. Identificación como usuario

Como hemos visto, las variables de sesión nos permiten mantener el estado en el servidor aunque el cliente cambie de página. Uno de los usos principales de las sesiones va a ser precisamente mantener el estado de que el usuario se ha autenticado con el sistema mediante un proceso tradicional de usuario/password de forma que pueda navegar por la parte segura del portal sin tener que identificarse nuevamente, además permitiendo que el servidor siempre sepa quién es el usuario.

En este apartado vamos a crear un sistema de usuarios. Para ello, comience creando un fichero de texto, preferiblemente en un directorio que no sirva el servidor web (no queremos que un desaprensivo se descargue nuestro fichero de passwords!). El formato del fichero va a ser texto con un línea para cada usuario y dos columnas, en la primera el identificador del

usuario (una palabra, sin espacios) y en el segundo su clave (de nuevo sin espacios).

Cree una página en la que aparezca un formulario en el que se le solicite al usuario su nombre y clave (escoja los campos de formulario más adecuados para cada caso). Al pulsar un botón del formulario se enviará el mismo a un script PHP que comprobará el nombre y clave contra el fichero antes definido. Si no se encuentra una línea con ese usuario y clave se saca una página indicando un error y se le ofrece al usuario volver a intentar el login. Si se encuentra una correspondencia se envía al usuario una página en la que en el título de la misma salga su nombre de usuario y como contenido el que deseen. Además, en dicha página habrá al menos un enlace. Dicho enlace llevará a otra página en la que el título es de nuevo el nombre del usuario y en el contenido de la misma se muestra el nombre y la password del usuario (para comprobar que mantienen la información del usuario sin que tenga que volver a autenticarse). Esta segunda página, si alguien intenta acceder a ella directamente, sin autenticarse previamente, debe redirigir al usuario a la página de login.



Esquema de las páginas

Checkpoint 2

Muestre estas páginas al profesor de prácticas

4. Trabajo opcional

En la primera página tras autenticarse el usuario añade la opción de que el usuario cambie su password.

5. Conclusiones

Hemos visto cómo crear scripts PHP que mantengan el estado de unos a otros, añadiendo así gran funcionalidad a páginas web básicas.

Depto. Automática y computación
Universidad Pública de Navarra

Daniel Morató
daniel.morato@unavarra.es