**TEMA 6: Sesiones, Seguridad y autenticación**

**1. Introducción**

Cualquier aplicación moderna tiene accesos restringidos, registros o diferenciación de roles por usuario. En PHP es muy sencillo aplicarlo usando sesiones y cookies.

Las sesiones controlan desde el servidor la información referente a un cliente, mientras que las cookies controlan dicha información desde el navegador. Ambas propiedades pertenecen al entorno de superglobals y nos permiten realizar un control de acceso y seguridad completo de una aplicación.

**2. Sesiones vs. Cookies**

En el caso de las sesiones, cuando un cliente realiza una petición, se inicia una nueva sesión en el servidor por cliente y dominio. Esta sesión se mantendrá abierta hasta que:

* El cliente cierre el navegador
* El servidor cierre la sesión por pasar el tiempo programado, porque el cliente la haya cerrado (logout) o por conveniencia propia de funcionamiento de la aplicación

En el caso de las cookies, cuando un cliente realiza la petición, se crea una cookie en el navegador que se esté utilizando. Al contrario que las sesiones, las cookies no se destruyen al cerrar el navegador, son permanentes y solo desaparecen cuando son destruidas.

**3. Superglobal**

Las variables predefinidas en PHP son superglobales y están disponibles en todos los ámbitos del script.

**4. Sesiones**

Las sesiones permiten almacenar datos de usuarios de forma individual, usando un ID de sesión único.

Normalmente un ID de sesión se envía al navegador mediante una cookie de sesión. De no existir un ID o Cookie de sesión, PHP creará una nueva sesión con un nuevo ID.

La sesión en PHP se activa usando **session\_start()**, y se podrá modificar usando el array **$\_SESSION**.

Las sesiones pueden ser destruidas mediante una función habilitada al usuario o por conveniencia de la aplicación.

**5. Control de acceso (PAGINA 16)**

El principal uso de las sesiones es integrarlas en el control de acceso de las aplicaciones.

**5.1. Objeto Seguridad**

El primer paso será crear una nueva clase que controle la sesión. Esta clase se localizaría en el Controlador del Patrón MVC.

Para controlar las sesiones, solo será necesario llamar a la clase seguridad allá donde sea requerida.

**5.2. El formulario de registro**

Una particularidad de los formularios de registro es que tienen un doble campo de contraseña, para verificar que el usuario no se ha equivocado.

Es importante que ambos sean de tipo **Password**, pero que el campo **name** sea distinto, para así poder compararlos posteriormente mediante el uso de $\_POST.

**5.3. Codificación de la contraseña**

Para codificar una contraseña se pueden utilizar funciones como sha o md5. Esta contraseña codificada será la que se almacene en base de datos.

**5.4. El formulario de acceso**

El formulario de acceso permite comprobar si un usuario ya está registrado en la base de datos con la misma password. Es importante en la comprobación tener en cuenta que la contraseña se guarda codificada.

**5.5. Activación de la sesión**

Una vez el usuario se ha registrado, se debe activar la sesión y añadir el usuario a la misma.

**5.6. Proteger una página**

Por lo tanto, para proteger una página:

* Se incluye la clase Seguridad y se crea el objeto.
* Se comprueba que exista el usuario.
* De no existir, se redirige a una zona de la aplicación no protegida.

**5.7. LogOut**

Para tener un logout, basta con crear una función que destruya la información de la sesión.

**6. Cookies**

La sesión guarda información individual del usuario mientras se mantiene abierta. En cambio, las cookies almacena datos en el navegador para identificar usuarios que regresan al sitio web. Esta información se guarda en el navegador del usuario durante un tiempo, incluso después de que se cierre.

Se trata de un mecanismo sencillo para almacenar información, pero puede ser alcanzada por otros softwares, así que no se debe almacenar información personal y sensible.

**6.1 Cookies: Creación y recuperación**

Las Cookies se crean y actualizan utilizando la función **setcookie()**. Desde PHP podemos decir el tiempo de duración de la cookie.