UNIDAD 4

# 1.- Autentificación de usuarios y control de acceso.

Los sitios web que necesitan emplear identificación del servidor, como las tiendas o los bancos, utilizan el protocolo **HTTPS**. Este protocolo requiere de un certificado válido, firmado por una **autoridad confiable**, que es verificado por el navegador cuando se accede al sitio web. Además, HTTPS utiliza métodos de cifrado para crear un canal seguro entre el navegador y el servidor, de tal forma que no se pueda interceptar la información que se transmite por el mismo.

Para identificar a los usuarios que visitan un sitio web, se pueden utilizar distintos métodos como el DNI digital o certificados digitales de usuario, pero el más extendido es solicitar al usuario cierta información que solo él conoce: la combinación de un nombre de usuario y una contraseña.

Anterior aprendiste a utilizar aplicaciones web para gestionar información almacenada en bases de datos. En la mayoría de los casos es importante implantar en este tipo de aplicaciones web, las que acceden a bases de datos, algún mecanismo de control de acceso que obligue al usuario a identificarse. Una vez identificado, se puede limitar el uso que puede hacer de la información.

Así, puede haber sitios web en los que los usuarios autentificados pueden utilizar sólo una parte de la información (como los bancos, que permiten a sus clientes acceder únicamente a la información relativa a sus cuentas). Otros sitios web necesitan separar en grupos a los usuarios autentificados, de tal forma que la información a la que accede un usuario depende del grupo en que éste se encuentre. Por ejemplo, una aplicación de gestión de una empresa puede tener un grupo de usuarios a los que permite visualizar la información, y otro grupo de usuarios que, además de visualizar la información, también la pueden modificar.

Debes distinguir la autentificación de los usuarios y el control de acceso, de la utilización de mecanismos para asegurar las comunicaciones entre el usuario del navegador y el servidor web. Aunque ambos aspectos suelen ir unidos, son independientes.

En los ejemplos de esta unidad, la información de autentificación (nombre y contraseña de los usuarios) se envía en texto plano desde el navegador hasta el servidor web. **Esta práctica es altamente insegura y nunca debe usarse sin un protocolo como HTTPS que permita cifrar las comunicaciones con el servidor web**. Sin embargo, la configuración de servidores web que permitan usar el protocolo HTTPS para cifrar la información que reciben y transmiten no forma parte de los contenidos de este módulo. Por este motivo, durante esta unidad utilizaremos únicamente el protocolo no seguro HTTP.

## 1.1.- Mecanismos de autentificación (I).

El protocolo HTTP ofrece un método sencillo para autentificar a los usuarios. El proceso es el siguiente:

* El servidor web debe proveer algún método para definir los usuarios que se utilizarán y cómo se pueden autentificar. Además, se tendrán que definir los recursos a los que se restringe el acceso y qué regla (ACL) se aplica a cada uno.
* Cuando un usuario no autentificado intenta acceder a un recurso restringido, el servidor web responde con un error de "Acceso no autorizado" (código 401).
* El navegador recibe el error y abre una ventana para solicitar al usuario que se autentifique mediante su nombre y contraseña.
* La información de autentificación del usuario se envía al servidor, que la verifica y decide si permite o no el acceso al recurso solicitado. Esta información se mantiene en el navegador para utilizarse en posteriores peticiones a ese servidor.
* Si las credenciales son erróneas **Apache** muestra "Acceso no autorizado" (código 401).

En el servidor web **Apache**, existe una utilidad en línea de comandos, **htpasswd**, que permite almacenar en un fichero una lista de usuarios y sus respectivas contraseñas. La información relativa a las contraseñas se almacena cifrada; aun así, es conveniente crear este fichero en un lugar no accesible por los usuarios del servidor web. Puedes tener información de este comando y sus opciones escribiendo en la terminal "**htpasswd --help**” o accediendo a la siguiente web: Utilidad en línea de comandos, htpasswd.

Por ejemplo, para crear el fichero de usuarios "proyecto.pass" y añadirle el usuario "gestor", puedes hacer:

htpasswd -c proyecto.pass gestor

e introducir la contraseña correspondiente a ese usuario.

La opción **"–c"** indica que se debe crear el fichero, por lo que solo deberás usarla cuando introduzcas el primer usuario y contraseña.

Para indicarle al servidor **Apache** qué recursos tienen acceso restringido, una opción es crear un fichero **.htaccess** en el directorio en que se encuentren, con las siguientes directivas:

AuthName "Contenido restringido"

AuthType Basic

AuthUserFile <ruta al fichero de usuarios>

require valid-user

El significado de cada una de las directivas anteriores es el siguiente:

**Directiva Significado**

**AuthName** Nombre de dominio que se usará en la autentificación. Si el cliente se autentifica correctamente, esa misma información de autentificación se utilizará automáticamente en el resto de las páginas del mismo dominio.

**AuthType** Método de autentificación que se usará. Además del método Basic, Apache también permite utilizar el método Digest.

**AuthUserFile** Ruta al archivo de credenciales que has creado con htpasswd.

**Require** Permite indicar que sólo puedan acceder algunos usuarios o grupos de usuarios concretos. Si indicamos "**valid-user**", podrán acceder todos los usuarios que se autentifiquen correctamente.

Además, tendrás que asegurarte de que en la configuración de Apache se utiliza la directiva **AllowOverride** para que se aplique correctamente la configuración que figura en los ficheros .htaccess.

Directiva AllowOverride: <https://httpd.apache.org/docs/2.0/es/mod/core.html#allowoverride>

## 1.2.- Mecanismos de autentificación (II).

Desde PHP puedes acceder a la información de autentificación HTTP que ha introducido el usuario utilizando el array "superglobal" **$\_SERVER**. Los valores se muestran en la tabla adjunta.

| **Valores del array $\_SERVER relacionados con la autentificación** | |
| --- | --- |
| **Valor** | **Contenido** |
| **$\_SERVER['PHP\_AUTH\_USER']** | Nombre de usuario que se ha introducido. |
| **$\_SERVER['PHP\_AUTH\_PW']** | Contraseña introducida. |
| **$\_SERVER['AUTH\_TYPE']** | Método HTTP usado para autentificar. Puede ser **Basic** o **Digest**. |

Es decir, que si creas una página web que muestre los valores de estas variables, y preparas el servidor web para utilizar autentificación HTTP, cuando accedas a esa página con el usuario "**gestor**" y contraseña "**secreto**" podrás observar los datos de la imagen izquierda de abajo. Si no introduces un usuario/contraseña válidos, el navegador te mostrará el error **401**.

Además, en PHP puedes usar la función **header** para forzar a que el servidor envíe un error de "Acceso no autorizado" (código **401**). De esta forma no es necesario utilizar ficheros **.htaccess** para indicarle a **Apache** qué recursos están restringidos. En su lugar, puedes añadir las siguientes líneas en tus páginas:

<?php

if (!isset($\_SERVER['PHP\_AUTH\_USER'])) {

header('WWW-Authenticate: Basic Realm="Contenido restringido"');

header('HTTP/1.0 401 Unauthorized');

echo "Usuario no reconocido!";

exit;

}

?>

La función **header** envía las [cabeceras HTTP](https://aulavirtual32.educa.madrid.org/ies.lapaz.alcobendas/pluginfile.php/66475/mod_resource/content/3/index.html#te82f626c-9b90-1636-11c0-1e703bd2d819), pero debe utilizarse antes de que se muestre nada por pantalla. En caso contrario, obtendrás un error.

[Manual online sobre la función **header**.](https://es.php.net/manual/es/function.header.php)

Con el código anterior, la página envía un error **401**, lo que fuerza al navegador a solicitar las credenciales de acceso (nombre de usuario y contraseña). Si se introducen, se ejecuta el resto de la página y se muestra su contenido. En este caso, **habría que añadir algún código para comprobar que el nombre de usuario y contraseña son válidos**. Si se pulsa el botón "Cancelar", se muestra el mensaje de error que se indica.

## 1.3.- Incorporación de métodos de autentificación a una aplicación web.

Si utilizas la función **header** para forzar al navegador a solicitar credenciales HTTP, el usuario introducirá un nombre y una contraseña. Pero el servidor no verificará esta información; deberás ser tú quien provea un método para comprobar que las credenciales que ha introducido el usuario son correctas.

El método más simple es incluir en el código PHP de tu página las sentencias necesarias para comparar los datos introducidos con otros datos fijos.

<?php

if ($\_SERVER['PHP\_AUTH\_USER']!='gestor' || $\_SERVER['PHP\_AUTH\_PW']!='secreto') {

header('WWW-Authenticate: Basic Realm="Contenido restringido"');

header('HTTP/1.0 401 Unauthorized');

echo "Usuario no reconocido!";

exit;

}

?>

Recuerda que el código PHP no se envía al navegador, por lo que la información de autentificación que introduzcas en el código no será visible por el usuario. Sin embargo, esto hará tu código menos portable. Si necesitas modificar el nombre de usuario o la contraseña, tendrás que hacer modificaciones en el código. Además, no podrás permitir al usuario introducir su propia contraseña.

Una solución mejor es utilizar un almacenamiento externo para los nombres de usuario y sus contraseñas. Para esto podrías emplear un fichero de texto, o mejor aún, una base de datos. La información de autentificación podrá estar aislada en su propia base de datos, o compartir espacio de almacenamiento con los datos que utilice tu aplicación web.

Si quieres almacenar la información de los usuarios en la base de datos "proyecto", tienes que crear una nueva tabla en su estructura. Para ello, revisa y ejecuta estas sentencias SQL. Para ello puedes usar **phpMyAdmin** o introducir el comando "**mysql -u root -p < archivo.sql"**. Donde archivo debe tener las sentencias siguientes:

Aunque se podrían almacenar las contraseñas en texto plano, por privacidad y seguridad es mejor hacerlo en formato encriptado. En el ejemplo anterior, para el usuario "admin" se almacena el  [**hash**](https://aulavirtual32.educa.madrid.org/ies.lapaz.alcobendas/pluginfile.php/66475/mod_resource/content/3/index.html#tb24f0301-d662-8fc8-bb3b-d44781324d44) (con el algoritmo **sha256**) correspondiente a la contraseña "secreto".

En PHP puedes usar la función **hash**, por ejemplo **hash('sha256', $cadena)** devuelve el **hash** utilizando el algoritmo **sha256**.

En **MySQL**  **select sha2('secreto', 256)** nos devolverá el **hash** (utilizando el mismo algoritmo **sha256**) de secreto.

Hay varios algoritmos para calcular el **hash** de una cadena. En nuestro proyecto debemos asegurarnos de usar el mismo tanto PHP como en **MySQL**. Por seguridad se recomienda no usar un algoritmo muy usado hasta ahora el **md5.** Encontrarás más información en el enlace siguiente: [Función hash.](https://www.php.net/manual/es/function.hash.php)