

# UD-05b: Seguridad y Control de Acceso

Desarrollo Web en Entorno Servidor

Curso 2020/2021

# Manejo de sesiones

## Inicio y fin de una sesión

Una forma de iniciar una sesión es ejecutar la función `session_start` para indicar a PHP que inicie una nueva sesión o reanude la anterior. Esta función devuelve *false* en caso de no poder *iniciar o restaurar* la sesión.

Como con *header*, hay que hacer las llamadas a esta función al principio del archivo, antes de que la página muestre información en el navegador.

La sesión puede almacenar información. Por tanto, todas las páginas que necesiten usar esta información, deberán ejecutar la función `session_start`. Mientras la sesión permanece abierta, podemos usar la variable superglobal `$_SESSION` para acceder o añadir a la información almacenada en la sesión.

# Manejo de sesiones

Por ejemplo, para contar el número de veces que el usuario visita la página:

```
<?php
// Iniciamos sesión o recuperamos la anterior sesión existente
session_start();
if (isset($_SESSION['visitas']))
    $_SESSION['visitas']++;
else
    $_SESSION['visitas'] = 1; // primera vez
?>
```

Si quisiéramos almacenar el instante en que se produce cada visita, la variable de sesión 'visitas' deberá ser un array, para almacenar cada uno de los instantes.

```
<?php
// Iniciamos sesión o recuperamos la anterior sesión existente
session_start();
// En cada visita añadimos un valor al array 'visitas'
$_SESSION['visitas'][] = time();
?>
```

# Manejo de sesiones

Si no cerramos la sesión de forma manual, los datos de una sesión se eliminan de forma automática pasado un tiempo, el cual se puede configurar en *php.ini*.

Existen dos funciones para eliminar manualmente la información de la sesión:

- **session\_destroy()** destruye la información asociada con la sesión actual. No destruye ninguna de las variables globales asociadas con la sesión, ni destruye la cookie de sesión. Para volver a utilizar las variables de sesión se debe llamar a *session\_start()*.
- **session\_unset()** libera todas las variables de sesión actualmente registradas.

# Manejo de sesiones

Ejemplo: Una BD llamada *bdsesiones* y una tabla llamada *usuarios*.





#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1 login 	varchar(15)	latin1_spanish_ci		No	Ninguna
<input type="checkbox"/>	2 password	binary(40)			No	Ninguna
<input type="checkbox"/>	3 nombre	varchar(25)	latin1_spanish_ci		No	Ninguna
<input type="checkbox"/>	4 lectura	tinyint(1)			No	1
<input type="checkbox"/>	5 escritura	tinyint(1)			No	0
<input type="checkbox"/>	6 administración	tinyint(1)			No	0

} tinyint, 1 byte [-128, 127]

**Metemos estos dos usuarios:**

```
INSERT INTO `usuarios` (`login`, `password`, `nombre`, `lectura`, `escritura`, `administracion`)
VALUES ('antonio', SHA1('123456'), 'antonio perez', '1', '0', '1'),
      ('paco', SHA1('78910'), 'paco perez', '1', '1', '0');
```

login	password	nombre	lectura	escritura	administracion
antonio	7c4a8d09ca3762af61e59520943dc26494f8941b	antonio perez	1	0	1
paco	66feeaf122435e47ed3015eb98b6917dbaf9a9f6	paco perez	1	1	0

5 elementos que están marcados:  Editar  Copiar  Borrar  Exportar

# Manejo de sesiones

login.php

```
<?php
session_start();
if (!empty($_POST['userid']) && !empty($_POST['password']))
{
    // El usuario acaba de intentar conectarse
    $userid = $_POST['userid'];
    $password = sha1($_POST['password']);
    // Conectamos con la Base de Datos y comprobamos la identidad del usuario
    $conexion = new mysqli('localhost','root','servidor1920','bdsesiones');
    $sql = "SELECT * FROM usuarios WHERE login='". $userid ." AND password='". $password ."'";
    $consulta = $conexion->query($sql);
    // Si existe un registro, el usuario ha proporcionado las credenciales correctas
    $resultado = $consulta->fetch_assoc();
    if ($resultado != null){
        // Guardamos los datos en la sesión
        $_SESSION['nombre'] = $resultado['nombre'];
        $_SESSION['lectura'] = $resultado['lectura'];
        $_SESSION['escritura'] = $resultado['escritura'];
        $_SESSION['administracion'] = $resultado['administracion'];
        header("Location:pagina.php");
    }
}
?>
```

Cuando comprobamos que las credenciales son correctas, la forma de crear una sesión es crear alguna variable de sesión. Así que, al menos la de `$_SESSION['nombre']` hay que crearla. En este ejemplo, además, se crean las otras para saber sus permisos, aunque esas podrían no estar.

Un `header("Location: ... ")` es un “salto” directo a la página que queramos. En este caso, cuando las credenciales del login son correctas y ya hemos guardado alguna variable de sesión, “saltamos” a la página “pagina.php”

```
<!DOCTYPE html>
<head>
    <meta charset="utf-8">
    <title>Desarrollo web en entorno servidor - ud 5</title>
</head>
<body>
    <h1>Pagina de entrada</h1>
    <?php
    if (isset($_SESSION['nombre'])) {
        // si existe usuario logueado, saltamos a pagina.php
        header("Location:pagina.php");
    }else{
        if (isset($userid)){
            // el usuario ha intentado conectarse y no lo ha conseguido
            echo 'No has conseguido acceder al sistema.<br>';
        }else{
            // el usuario todavía no ha intentado autenticarse
            echo 'Loguéate para entrar en el sistema.<br>';
        }
    }
    <!-- si no hay ningún usuario logueado, mostramos el formulario para loguearse. -->
    <form name="formulariologin" method="POST" action="login.php">
    <label for="userid">Usuario</label><input type="text" name="userid"><br>
    <label for="password">Contraseña</label><input type="password" name="password"><br>
    <input type="submit" name="entrar" value="Entrar">
    </form>
    <?php
    }
    ?>
</body>
</html>
```

Faltarían las comprobaciones de que se han de introducir los dos campos.

pagina.php

Aquí llegamos tras un login correcto. Es la página principal, donde el usuario ya podría hacer cosas.

```
<?php
    session_start(); // Permite continuar la sesión.
    // Si no existe la variable de sesión, volvemos a login.php
    if (!isset ($_SESSION['nombre'])){
        header("Location:login.php");
    }
    echo "¡Bienvenido, ". $_SESSION['nombre'] ."!<br>";
    echo "<a href='logout.php'>Cerrar Sesión</a>";
?>
<!DOCTYPE html>
<head>
    <meta charset="utf-8">
    <title>Desarrollo web en entorno servidor - ud 5</title>
</head>
<body>
    <h1>Permisos de acceso en esta aplicación:</h1>
    <ul>
        <?php
            if ($_SESSION['lectura'] == 1)
                echo '<li> Lectura </li>';
            if ($_SESSION['escritura'] == 1)
                echo '<li> Escritura </li>';
            if ($_SESSION['administracion'] == 1)
                echo '<li> Administracion </li>';
        ?>
    </ul>
</body>
</html>
```

Esto lo hacemos para que un usuario NO LOGUEADO no se pueda colar escribiendo en el navegador: *pagina.php*. Comprobamos si está logueado, revisando que si no existe la variable de sesión nombre, se le envía a *login.php*

Aquí debajo sólo se le informa al usuario de los permisos que tiene. En una página más elaborada, podría haber una tabla para ver y añadir productos, o gestionar los permisos de los usuarios mostrando u ocultando botones, por ejemplo.



## logout.php

```
<?php
session_start(); // Permite continuar la sesión.
if (isset ($_SESSION['nombre']))
{
    $_SESSION = array();
    session_destroy();
}
header("Location:login.php");
?>
```

Comprobamos si hay sesión iniciada, en cuyo caso borramos las variables de sesión y eliminamos la sesión. Finalmente, siempre redirigimos a la página login.php

# Cookies

- Una **cookie** es un fichero de texto que se guarda en el navegador del usuario.
- Su uso más típico es el almacenamiento de las preferencias del usuario, como por ejemplo, el idioma en que se deben mostrar las páginas, para así no tener que volver a indicarlo la próxima vez que visite el sitio.

Para almacenar una cookie en el navegador, podemos utilizar la función `setcookie`. El único parámetro obligatorio que tenemos que usar es el nombre de la cookie, pero admite varios parámetros más opcionales.

<http://es.php.net/manual/es/function.setcookie.php>

```
setcookie("nombre_usuario", $_SERVER['PHP_AUTH_USER'], time()+3600);
```

Nombre de  
la cookie

Valor de  
la cookie

Fecha caducidad en segundos (1 h. después).  
*Si no se especifica este parámetro, la cookie  
se elimina al cerrar el navegador.*



**Ojo, es sólo un ejemplo, no es aconsejable almacenar en las cookies información relativa a la seguridad.**

# Cookies

- Las sentencias `setcookie(...)` también deben usarse antes de que el navegador muestre información en pantalla. Se debe a que las cookies también se transmiten entre el navegador y el servidor utilizando los encabezados del protocolo HTTP.
- Para recuperar la información almacenada en una cookie, basta con acceder al array `$_COOKIE`. Es el navegador quien recupera automáticamente en el array toda la información de las cookies de ese sitio en concreto.
- Al usar cookies en un sitio web, tendremos en cuenta que su disponibilidad está controlada por el cliente. Algunos usuarios deshabilitan las cookies en el navegador por motivos de seguridad, o simplemente, el usuario puede eliminarlas manualmente de su sistema.
- Podemos eliminar una cookie antes de que su tiempo expire, con la misma función `setcookie` pero indicando una fecha de caducidad anterior a la actual.


Este ejemplo es igual que el anterior, pero aquí hemos guardado nueva información en una cookie.

<?php

```
if(!isset($_SERVER['PHP_AUTH_USER'])) {
    // si el usuario no está autenticado, pedimos credenciales.
    header('WWW-Authenticate: Basic realm="Contenido restringido"');
    header("HTTP/1.0 401 Unauthorized");
    exit;
}else{
    @$conexion = new mysqli("localhost", "alumno", "12345", "bdprueba");
    if ($conexion->connect_errno == null) {
        // como trabajamos con time(), es conveniente ajustarlo bien.
        date_default_timezone_set('Europe/Madrid');
        $sql = "SELECT usuario FROM usuarios WHERE usuario = '".$_SERVER['PHP_AUTH_USER']."'
                AND contraseña = md5('".$_SERVER['PHP_AUTH_PW']."'");
        $consulta = $conexion->query($sql);
        if($consulta->fetch_assoc() == null) {
            // Si no existe, se vuelven a pedir las credenciales
            header('WWW-Authenticate: Basic realm="Contenido restringido"');
            header("HTTP/1.0 401 Unauthorized");
            exit;
        } else {
            if (isset($_COOKIE['hora_ultimo_logueo'])) {
                $ultimo_login = $_COOKIE['hora_ultimo_logueo'];
            }
            setcookie("hora_ultimo_logueo", time(), time()+60*60*24*365);
        }
        $consulta->close();
        $conexion->close();
    }
}
```

?>





```
<!DOCTYPE html>
<head>
    <meta charset="utf-8">
    <title>Desarrollo web en entorno servidor - ud 5</title>
</head>
<body>

    <?php

        echo "Nombre de usuario: ".$_SERVER['PHP_AUTH_USER']."<br />";
        echo "Contraseña: ".$_SERVER['PHP_AUTH_PW']."<br />";
        echo "Hash de la contraseña: ".md5($_SERVER['PHP_AUTH_PW'])."<br />";

        if (isset($hora_ultimo_login)) {
            echo "Hora de último login: " . date("d/m/y \a \l\a\s H:i", $hora_ultimo_login);
        }else{
            echo "¡Bienvenido por primera vez!";
        }

    ?>

</body>
</html>
```



# EJERCICIO

Crea una carpeta de nombre **ud05ejer02** que incluya los siguientes archivos:

- ✓ registro.php
- ✓ login.php
- ✓ pagina.php
- ✓ logout.php

Usa la misma tabla **usuarios** de la BD **bdsesiones** del ejemplo anterior:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1 login	varchar(15)	latin1_spanish_ci		No	Ninguna
<input type="checkbox"/>	2 password	binary(40)			No	Ninguna
<input type="checkbox"/>	3 nombre	varchar(25)	latin1_spanish_ci		No	Ninguna
<input type="checkbox"/>	4 lectura	tinyint(1)			No	1
<input type="checkbox"/>	5 escritura	tinyint(1)			No	0
<input type="checkbox"/>	6 administración	tinyint(1)			No	0

Crea un formulario para añadir usuarios a la tabla.  
Comprueba que las contraseñas son idénticas. Usa el algoritmo SHA1 al insertarlas.  
Inserta 3 usuarios llamados **aaa**, **bbb** y **ccc**, con las contraseñas que prefieras.

## Registrarme

Usuario:

Password:

Repite Pass:

¡Debe introducir un password!

¡Debe introducir un password!

## Registrarme

Usuario:

Password:

Repite Pass:

¡Debe introducir un nombre de usuario!

Ahora, desde PHPMyAdmin, edita el campo *escritura* del usuario **bbb** y los campos *escritura* y *administracion* de **ccc**. Debe quedar así:

		usuario	contrasena	lectura	escritura	administracion
<input type="checkbox"/>	Editar	aaa	7e240de74fb1ed08fa08d38063f6a6a91462a815	1	0	0
<input type="checkbox"/>	Editar	bbb	5cb138284d431abd6a053a56625ec088bfb88912	1	1	0
<input type="checkbox"/>	Editar	ccc	f36b4825e5db2cf7dd2d2593b3f5c24c0311d8b2	1	1	1

La web de login tendrá este aspecto, y si intentamos entrar sin completar alguno de los campos, marcará el error:

# LOGIN

Introduce tus credenciales para entrar

Usuario

Contraseña  <-- ¡Debes introducir tu password!

¿Aún no te has registrado? [¡Regístrate!](#)

Debe existir esa pareja usuario/contraseña.

# LOGIN

Datos incorrectos. Prueba de nuevo.

Usuario

Contraseña

¿Aún no te has registrado? [¡Regístrate!](#)

Al clicar en [¡Regístrate!](#), nos mandará a la página [registro.php](#) y allí, tras terminar el INSERT, nos redireccionará a [login.php](#) (usando *header*, como hace el *logout*).

Al principio se da la bienvenida al usuario, se le dicen las acciones que puede realizar (que para **ccc** son: *ver*, *añadir* y *eliminar*) y se le muestra un enlace para cerrar sesión (logout). Después, se muestra el código del sitio web de **libros**.

¡Bienvenido, **ccc**! En esta página puedes: (ver) (añadir) (eliminar) libros [[logout](#)]

Tu última conexión fue el [día a las hora].

## Tabla Libros

Id:

Título:

Autor:

Páginas:

Haz uso de cookies  
para mostrar este dato.

Al clicar en [logout](#), cerramos la sesión y  
volvemos a la pantalla inicial para hacer login.

ID	TITULO	AUTOR	PAGINAS	ELIMINAR
1	aaa	jajaja	111	<a href="#">Eliminar</a>
2	eee	jejeje	222	<a href="#">Eliminar</a>
3	iii	jijiji	333	<a href="#">Eliminar</a>
4	ooo	jojojo	444	<a href="#">Eliminar</a>
5	uuu	jujuju	555	<a href="#">Eliminar</a>



Sin embargo, el usuario **bbb** verá esto:

¡Bienvenido, **bbb**! En esta página puedes: (ver) (añadir) libros [[logout](#)]  
Tu última conexión fue el [día a las hora].

## Tabla Libros

Id:

Título:

Autor:

Páginas:

ID	TITULO	AUTOR	PAGINAS
1	aaa	jajaja	111
2	eee	jejeje	222
3	iii	jijiji	333
4	ooo	jojojo	444
5	uuu	jujuju	555

Fíjate en que ya no aparece (*eliminar*) en sus permisos. Además, la tabla ya no muestra la columna (ELIMINAR). Esto se debe a que el usuario **bbb** tiene un 0 en *administracion*.

Ahora falta ver qué pasa con **aaa**, quien verá lo siguiente:

¡Bienvenido, **aaa**! En esta página puedes: (ver) libros [[logout](#)]  
Tu última conexión fue el [día a las hora].

## Tabla Libros

Id:   
Título:   
Autor:   
Páginas:

ID	TITULO	AUTOR	PAGINAS
1	aaa	jajaja	111
2	eee	jejeje	222
3	iii	jijiji	333
4	ooo	jojojo	444
5	uuu	jujuju	555

Sólo aparece el permiso (ver).  
Por tanto, tiene todos los campos del formulario bloqueados (con *disabled*).  
Esto es por su 0 en escritura.