

# Índice: Tema 4.3

## **4.3 Aplicaciones basadas en Sesiones**

4.3.1 Manejo de Cookies

4.3.2 Manejo de Sesiones

4.3.3 Aplicación basada en sesiones

# Índice: Tema 4.3

## **4.3 Aplicaciones basadas en Sesiones**

### **4.3.1 Manejo de Cookies**

### 4.3.2 Manejo de Sesiones

### 4.3.3 Aplicación basada en sesiones



## 4.3 APLICACIONES BASADAS EN SESIONES

### 4.3.1 MANEJO DE COOKIES



#### ➤ Sesiones y cookies

- ✓ Las **sesiones** y las **cookies** proporcionan la habilidad de "recordar" información sobre los usuarios.
- ✓ Se diferencian en cómo son capaces de almacenar dicha información:
  - Las sesiones almacenan los datos en archivos temporales en el disco duro del servidor.
  - Las cookies son pequeños archivos que se almacenan en la máquina del cliente.
- ✓ Las cookies son útiles para los programadores ya que permiten el poder compartir variables que se necesitan en múltiples páginas de una manera fácil y segura.
- ✓ Las cookies son simples pares cadenas de texto "**nombre=valor**", que tienen asociada una URL.
- ✓ El navegador usa esta URL para decidir si envía o no la cookie al servidor.
- ✓ Los navegadores WEB tienen control sobre las cookies pudiéndose desactivar.



#### ➤ Características de las cookies

- ✓ Existen dos tipos de cookies:
  - Persistentes: La cookie es almacenada por el navegador.
  - Las cookies persistentes están mal vistas por los usuarios.
  - No Persistentes: Los valores de la cookie se pierden al cerrar el navegador.
- ✓ Las cookies no pueden acceder a ninguna información sobre el sistema del usuario.
- ✓ Las cookies están restringidas a un cierto ámbito o rango de direcciones, en el cual puede ser usada:
  - El programador define este ámbito.
  - El navegador lee la información sobre este ámbito y determina si un servidor tiene acceso a dicha cookie.



## 4.3 APLICACIONES BASADAS EN SESIONES

### 4.3.1 MANEJO DE COOKIES



#### ➤ Generación y recuperación de cookies (1)

- ✓ Aunque las cookies son almacenadas en el equipo del cliente, se generan desde un programa en el servidor (por ejemplo en PHP).
- ✓ Cuando se realiza una transferencia HTTP entre cliente y servidor, las cookies son incluidas en la cabecera HTTP, por lo que cualquier orden de generación de un cookie desde una página web debe ser dada antes de cualquier comando HTML.
- ✓ Para establecer una cookie se dispone de la función `setcookie(nombre, valor)`. Ejemplo: `<?php setcookie("nombre", "Alberto"); ?>`
  - Esta función le dice a un navegador que recuerde el nombre y valor de una cookie y lo envíe de vuelta al servidor en peticiones posteriores.
  - Devuelve un valor numérico indicando si se produjo algún error en el proceso de creación de la cookie. Un valor distinto de cero indica que el proceso se realizó correctamente.
- ✓ Para borrar la cookie basta con omitir su valor `setcookie(nombre)`.



#### ➤ Generación y recuperación de cookies (2)

- ✓ La sintaxis completa de la función `setcookie()` es:

```
int setcookie(string name [,string value] [,int expire] [,string path]
              [,string domain] [,bool secure] [,bool httponly])
```

- **name** Nombre de la cookie a establecer.
- **value** Valor de la cookie a establecer.
- **expire** Tiempo de vida a partir del cual los datos de la cookie no serán válidos.
- **path** Ruta desde la raíz del servidor a partir de la cual la cookie es accesible.
- **domain** El dominio a partir del cual la cookie es accesible.
- **secure** Especifica si la cookie sólo debe ser transmitida a través de una conexión HTTPS segura. El valor por defecto es FALSE.
- **httponly** Este valor indica si la cookie es accesible fuera de una petición HTTPS. Por ejemplo por lenguajes de scripting, como JavaScript. El valor por defecto es FALSE.



#### ➤ Generación y recuperación de cookies (3)

- ✓ Una vez la cookie ha sido generada, esta permanece en el equipo del cliente y será enviada al servidor cada vez que el cliente realicen consultas al mismo.
- ✓ Los valores almacenados en las cookies que llegan al servidor son accesibles desde PHP desde el array superglobal `$_COOKIE`.

✓ Ejemplo:

```
<?php setcookie("nombre","Alberto"); ?>
<html>
  <head><title>Ejemplo</title>
</head>
  <body>
    <?php echo "Te llamas ".$_COOKIE['nombre']; ?>
  </body>
</html>
```

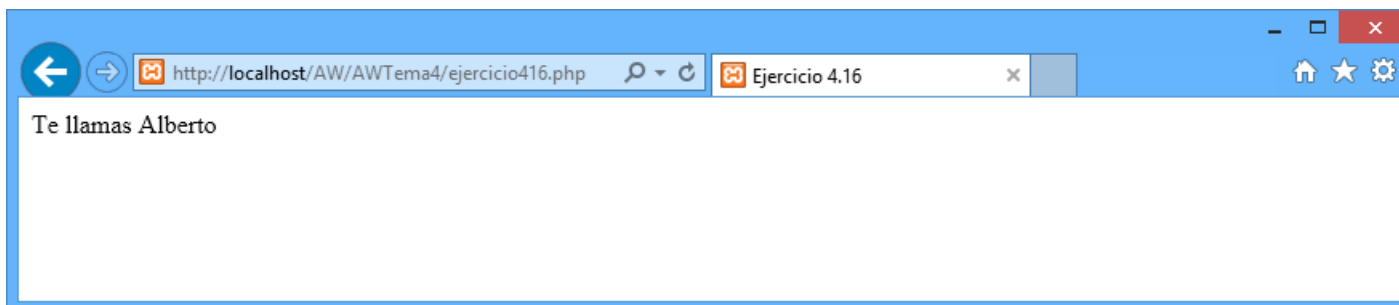


#### ➤ Generación y recuperación de cookies (4)

- ✓ En el primer acceso, como el cliente aún no tiene la cookie no la puede enviar y el servidor no la puede utilizar para construir la página solicitada:



- ✓ En un segundo acceso, el cliente ya dispone de la cookie (enviada por el servidor en respuesta al primer acceso) y la envía al servidor para que este pueda construir la página solicitada:





#### ➤ Generación y recuperación de cookies (5)

- ✓ Hacer una página en PHP que genere una cookie con el nombre introducido mediante un formulario para poder utilizarlo como valor por defecto al solicitar la página de nuevo en una petición posterior.

Introducimos el nombre y pulsamos Enviar.

La propia página procesa los datos en el servidor y genera la cookie que se envía al cliente.

Al solicitar de nuevo la página en otra pestaña, la cookie enviada en la solicitud es utilizada por el servidor para completar el campo.



#### ➤ Expiración de cookies

- ✓ Si omitimos este parámetro la cookie desaparecerá al cerrar el cliente el navegador.
- ✓ El tiempo de vida de una cookie es relativo al número de segundos que han pasado desde el 1 de Enero de 1970. Por tanto el tiempo de expiración de la cookie tiene que ser relativo a esta fecha.
- ✓ PHP nos proporciona un conjunto extenso de funciones para el tratamiento de fechas. En particular `time()` nos proporciona el número de segundos que han transcurrido desde 1 de Enero de 1970 hasta la fecha y momento en el que se realiza la llamada a dicha función.

```
// Esta cookie caduca en una hora
```

```
setcookie("mi_cookie", $value, time() + 60*60);
```

```
// Esta cookie caduca a las 0h:00m:00s del 20 de Mayo de 2006
```

```
setcookie("mi_cookie", $value, mktime(0,0,0,05,20,2006));
```

```
// Borrado de una cookie con tiempo de vida establecido
```

```
setcookie("mi_cookie", "", time());
```



## 4.3 APLICACIONES BASADAS EN SESIONES

### 4.3.1 MANEJO DE COOKIES



#### ➤ **Ámbito de las cookies**

- ✓ Con los parámetros **ruta** y **dominio** se controla el ámbito en el que puede ser accesible una cookie.
- ✓ La ruta de acceso por defecto de una cookie es **/** lo que quiere decir que la cookie es accesible por cualquier subdirectorío del raíz.
- ✓ Para limitar el acceso a la cookie a un subdirectorío tenemos que usar como parámetro la ruta a dicho subdirectorío:
  - Por ejemplo, si establecemos **/sitio/** como parámetro, la cookie podrá ser accesible por todas las páginas alojadas a partir de dicho directorío.
  - Cuidado con poner **/sitio**, la cookie podría ser accesible por la página **/sitio.php** o **/sitio1/pagina.php**, etc...
- ✓ Para limitar el acceso a la cookie a un único archivo tenemos que establecer como parámetro la ruta al archivo. Por ejemplo **/sitio/pagina.php**.



#### ➤ Expiración y ámbito de las cookies (1)

- ✓ Utilice las cookies para generar un contador de visitas de las páginas enlazadas desde la web “ejercicio17.php” que se resetee si transcurren más de 5 segundos sin visitar la página. Además, dichas páginas hacen uso del nombre introducido en el formulario mediante una cookie que tan sólo es visible en “pagina1.php”.

# Índice: Tema 4.3

## **4.3 Aplicaciones basadas en Sesiones**

4.3.1 Manejo de Cookies

**4.3.2 Manejo de Sesiones**

4.3.3 Aplicación basada en sesiones



#### ➤ Sesiones y cookies

- ✓ Como se ha comentado en el apartado anterior, las sesiones son otra alternativa a las cookies para “recordar” cierta información suministrada por el usuario cuando navega a través de las diferentes páginas de un sitio web.
- ✓ Al contrario de las cookies, las variables de sesión se almacenan en el servidor y desaparecen (a no ser que se cambie la configuración del servidor) cuando el cliente cierra el navegador. Funciones de utilidad:
  - `session_start()` Nueva sesión o sesión abierta. Se debe colocar al comienzo del script de cualquier página que utilice la sesión.
  - `session_id()` Devuelve el identificador de la sesión.
  - `session_destroy()` Finaliza de la sesión.
- ✓ Todas las variables a almacenar en la sesión se guardan en el array superglobal `$_SESSION`.



#### ➤ Probando las variables de sesión (1)

- ✓ Usar las variables de sesión para almacenar el número de visitas a la web durante la sesión y el tiempo de permanencia en la página:

```
<?php
    session_start();
    if(!isset($_SESSION["contador"])) {
        $_SESSION["contador"] = 0;
        $_SESSION["inicio"] = time();
    }
    $_SESSION["contador"]++;
?>
<html>
  <head><title>Ejercicio 17</title></head>
  <body>
    <p>Esta sesión lleva = <?php echo "ID de sesión = ".session_id() ?></p>
    <p>Cuenta = <?php echo $_SESSION["contador"] ?></p>
    <p>Inicio = <?php echo $_SESSION["inicio"] ?></p>
    <p>La sesión lleva <?php echo time()-$_SESSION["inicio"] ?> segundos</p>
  </body>
</html>
```



#### ➤ Probando las variables de sesión (2)

✓ Resultado:

The image displays three sequential browser screenshots of a web application at the URL `http://localhost/GIE/AW/Tema4/Ejercicio17/ejercicio17.php`. Each screenshot shows the same session ID: `fhivkf534k98vi9bq0i5fjkjg7`. The variables shown in each screenshot are:

- First screenshot:** Cuenta = 1, Inicio = 1357924671, Esta sesión lleva 0 segundos.
- Second screenshot:** Cuenta = 2, Inicio = 1357924671, Esta sesión lleva 15 segundos.
- Third screenshot:** Cuenta = 3, Inicio = 1357924671, Esta sesión lleva 24 segundos.



# Índice: Tema 4.3

## **4.3 Aplicaciones basadas en Sesiones**

4.3.1 Manejo de Cookies

4.3.2 Manejo de Sesiones

**4.3.3 Aplicación basada en sesiones**



#### ➤ Aplicación basada en sesiones (1)

- ✓ Para mostrar la utilidad de las variables de sesión, se va a diseñar una aplicación que de acceso a una web mediante un usuario y contraseña. Para ello:
  - La página de entrada (index.php) recoge los datos acceso y los envía a otra (login.php) con los que se abre una sesión.
  - Mientras la sesión esté abierta, el usuario puede acceder a la página de bienvenida (welcome.php) y otras del mismo sitio web (App. pages).
  - La página de salida (logout.php) destruye la sesión haciendo que las páginas del sitio ya no sean accesibles a menos que se ingrese de nuevo.





#### ➤ Página principal (1): ejercicio20.php

- ✓ La página principal “ejercicio20.php” proporciona el formulario de entrada a la aplicación y los datos se envían a la página “login.php” para su procesamiento:

```
<html>
  <head>
    <title>Ejercicio 20</title>
    <link href="css/estilos.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <h1>Aplicación basada en Sesiones</h1><hr/>
    <p>Por favor, introduzca los datos de acceso:</p>
    <form name="formulario" action="login.php" method="post">
      <label for="usuario">Usuario</label>
      <input type="text" name="usuario" id="usuario" />
      <label for="clave">Contraseña</label>
      <input type="password" name="clave" id="clave"/>
      <input type="submit" value="Entrar"/>
    </form>
  </body>
</html>
```



## 4.3 APLICACIONES BASADAS EN SESIONES

### 4.3.3 APLICACIÓN BASADA EN SESIONES

### EJERCICIO 20



#### ➤ Página principal (2): ejercicio20.php

- ✓ Los datos de acceso a la web son, usuario=**alberto** y contraseña=**1234**:

The screenshot shows a web browser window with the address bar containing `http://localhost/GIE/AW/Tema4/Ejercicio18/ejercicio18.php`. The page title is "Ejercicio 18". The main content of the page is a login form titled "Aplicación basada en Sesiones". Below the title, there is a prompt: "Por favor, introduzca los datos de acceso:". The form consists of two input fields: "Usuario" with the value "alberto" and "Contraseña" with masked characters "•••••". Below the input fields is a button labeled "Entrar". The status bar at the bottom of the browser shows the URL `http://localhost/GIE/AW/Tema4/Ejercicio18/setup.php` and a zoom level of 100%.



#### ➤ Página de entrada (1): login.php

- ✓ La página “login.php” comprueba si el usuario y contraseña son correctos. En caso afirmativo, se crea una variable de sesión con el nombre del usuario y se accede a la página “welcome.php”, en caso negativo se deniega el acceso:

```
<?php
    if (($_POST["usuario"]=="alberto") && ($_POST["clave"]=="1234")) {
        session_start();
        $_SESSION["nombre"] = $_POST["usuario"];
        header("Location: welcome.php");
    }
?>
<html>
    <head>
        <title>Página de entrada</title>
        <link href="css/estilos.css" rel="stylesheet" type="text/css"/>
    </head>
    <body>
        <h1>Aplicación basada en Sesiones</h1><hr/>
        <p>No está autorizado para utilizar esta aplicación.</p>
    </body>
</html>
```



## 4.3 APLICACIONES BASADAS EN SESIONES

### 4.3.3 APLICACIÓN BASADA EN SESIONES

### EJERCICIO 20



#### ➤ **Página de entrada (2): login.php**

- ✓ Resultado de introducir un usuario y/o contraseña incorrectos:





#### ➤ Página de bienvenida (1): welcome.php

- ✓ La página “welcome.php” comprueba si el usuario tiene la sesión abierta para mostrar la página, en caso negativo redirecciona a la página de entrada :

```
<?php
    session_start();
    if(!isset($_SESSION["nombre"]))
        header("Location: ejercicio20.php");
?>
<html>
    <head>
        <title>Página de bienvenida</title>
        <link href="css/estilos.css" rel="stylesheet" type="text/css"/>
    </head>
    <body>
        <h1>Aplicación basada en Sesiones</h1><hr/>
        <p>Hola <?php echo $_SESSION["nombre"] ?>, bienvenido a la aplicación.</p>
        
        <input type="button" value="Salir"
            onclick="window.location.href='logout.php' " />
    </body>
</html>
```



## 4.3 APLICACIONES BASADAS EN SESIONES

### 4.3.3 APLICACIÓN BASADA EN SESIONES

### EJERCICIO 20



#### ➤ Página de bienvenida (2): welcome.php

- ✓ Resultado de tener la sesión abierta:







#### ➤ Página de salida(1): logout.php

- ✓ La página “logout.php” elimina las variables de sesión, y muestra la página de salida en la que se da la posibilidad de volver a ingresar redireccionando a la página de inicio mediante un botón:

```
<?php
    session_start();
    session_destroy();
?>
<html>
    <head>
        <title>Página de salida</title>
        <link href="css/estilos.css" rel="stylesheet" type="text/css"/>
    </head>
    <body>
        <h1>Aplicación basada en Sesiones</h1><hr/>
        <p>Gracias por utilizar la aplicación.</p>
        <input type="submit" id="boton" value="Volver a entrar"
            onclick="window.location.href='ejercicio20.php'" />
    </form>
</html>
```



#### ➤ **Página de salida(2): logout.php**

- ✓ Resultado de pulsar salir en la página “welcome.php”:

