

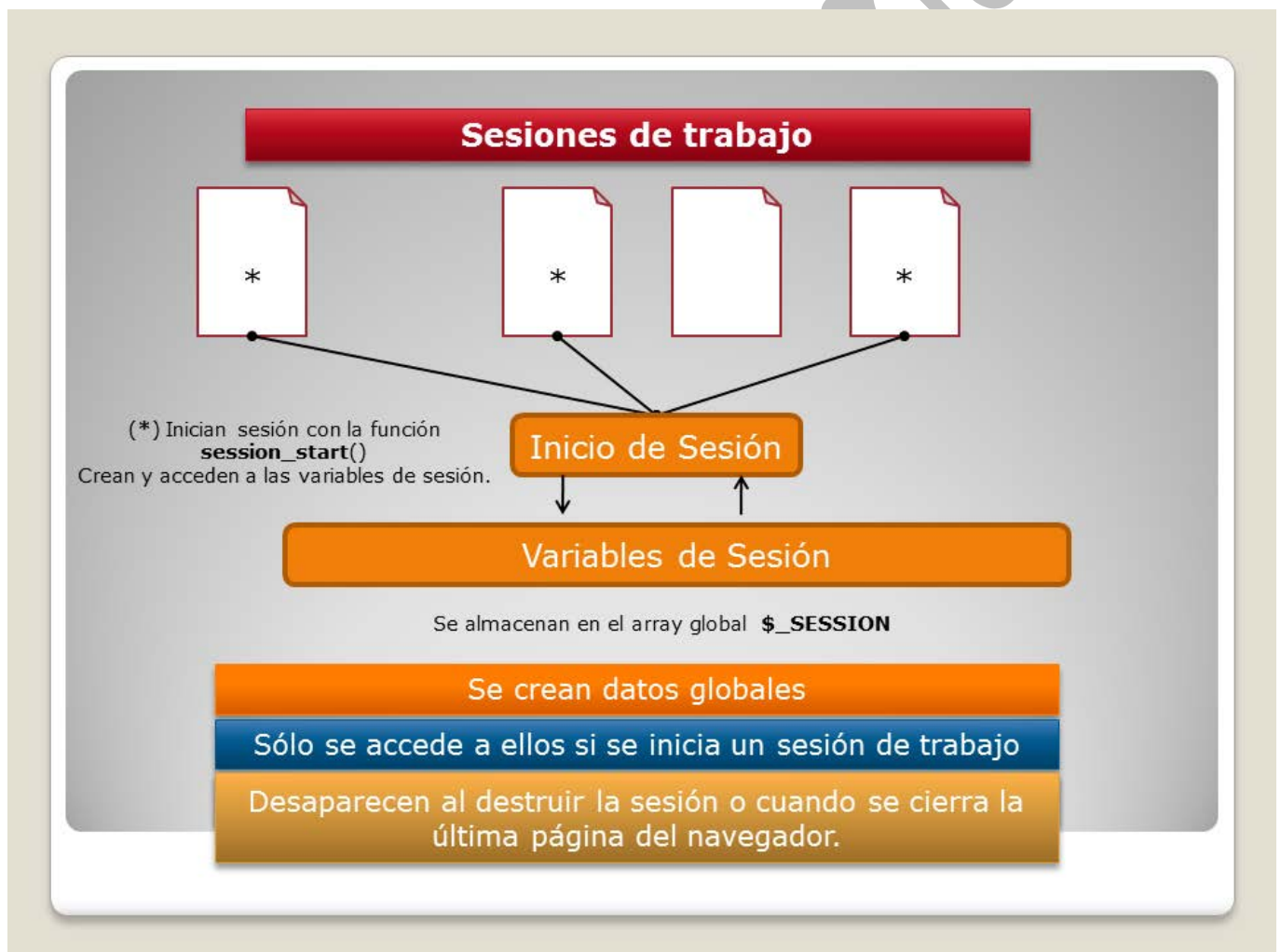
# Sesiones de trabajo

Las sesiones, en aplicaciones web realizadas con PHP y en el desarrollo de páginas web en general, nos sirven para almacenar información que se memorizará durante toda la visita de un usuario a una página web y que se mantienen durante un periodo de tiempo determinado o hasta que se cierra el navegador.

Dicho de otra forma, un usuario puede ver varias páginas durante su paso por un sitio web y con sesiones podemos almacenar variables que podremos acceder en cualquiera de esas páginas.

Digamos que las sesiones son una manera de guardar información, específica para cada usuario, durante toda su visita. Cada usuario que entra en un sitio abre una sesión, que es independiente de la sesión de otros usuarios. En la sesión de un usuario podemos almacenar todo tipo de datos, como su nombre, productos de un hipotético carrito de la compra, preferencias de visualización o trabajo, páginas por las que ha pasado, etc. Todas estas informaciones se guardan en lo que denominamos variables de sesión.

Todas las páginas que necesiten acceder a la información almacenada en las variables de sesión, tendrán que iniciar una sesión de trabajo para conectarse a la sesión abierta.



## Iniciar Sesión

Cuando queremos utilizar variables de sesión en una página tenemos que iniciar la sesión con la función:

```
bool session_start()
```

Inicia una sesión para el usuario o continúa la sesión que pudiera tener abierta en otras páginas. Al hacer `session_start()` PHP internamente recibe el identificador de sesión almacenado. Si no existe tal identificador se sesión, simplemente lo crea.

Una vez inicializada la sesión con `session_start()` podemos utilizar variables de sesión, es decir, almacenar datos para ese usuario, que se conserven durante toda su visita o recuperar datos almacenados en páginas que haya podido visitar.

La sesión se tiene que inicializar antes de escribir cualquier texto en la página. Esto es importante y de no hacerlo así corremos el riesgo de recibir un error, porque al iniciar la sesión se deben leer las cookies del usuario, algo que no se puede hacer si ya se han enviado las cabeceras del HTTP, como ya habíamos comentando en el apartado de trabajar con Cookies.

## Acceder a los datos de la sesión

Una vez iniciada la sesión podemos utilizar variables de sesión a través de `$_SESSION`, que es un array asociativo, donde se accede a cada variable a partir de su nombre:

```
$_SESSION["nombre_de_variable"]
```

Toda página que haya iniciado sesión, tendrá acceso a la información almacenada en este array con el que se puede compartir información entre páginas.

Una de las opciones de trabajo interesantes con las sesiones es la posibilidad de impedir el acceso a ciertas páginas de la web si no se ha iniciado una sesión por medio de un nombre de usuario y contraseña válidos.

## Cerrar Sesión

Cuando un usuario se desconecta de nuestra página, lo más normal es cerrar la sesión abierta y destruir todos los datos de trabajo que se hayan creado.

Para destruir la información generada durante una sesión se utiliza la función `session_destroy()`.

```
bool session_destroy(void)
```

Esta función aunque destruye la información generada en la sesión, no elimina las variables globales creadas durante la sesión y tampoco la cookie de sesión del usuario.

Si iniciáramos de nuevo la sesión con `session_start()`, volveríamos a utilizar las variables de sesión creadas durante la sesión anterior.

Para cerrar y destruir completamente la sesión tendremos que eliminar las variables y la id de sesión.

Para destruir las variables generadas durante la sesión se utiliza la función `session_unset()`.

```
bool session_unset(void)
```

Para destruir el identificador de sesión del usuario si se han utilizado cookies, tendríamos que volver a definir la cookie con un tiempo de vida negativo, como se muestra a continuación:

```
setcookie(session_name(),'',-4200,'/');
```

Por lo tanto si deseamos hacer bien las cosas el proceso implicaría: eliminar la cookie si existe, destruir las variables y cerrar la sesión.

A continuación os muestro una posible solución por medio de una función que se encargue de realizar todas las operaciones:

```
function CerrarSesion() {
// Destruimos la cookie de sesión si existe
if(isset($_COOKIE[session_name()])) {
    setcookie(session_name(),'',-4200,'/');
}
session_unset(); // Destruimos las variables de sesión
session_destroy(); // Destruimos finalmente la información de la sesión
}
```

## Ejemplo

Veamos un ejemplo sencillo en el que por medio de cuatro páginas se muestren los pasos de crear las variables, consultarlas y la destrucción de la sesión iniciada.

- *inicio.php*: muestra tres enlaces, uno a la página que crea las variables de sesión y otro a la página que comprueba si existen las variables y el último a la página que elimina las variables.
- *crea\_valores.php*: inicia una sesión y crea dos variables.
- *comprueba\_valores.php*: inicia sesión y comprueba si existen las variables de sesión mostrando su contenido.
- *destruye\_valores.php*: cierra la sesión abierta y destruye las variables de sesión creadas.

### inicio.php

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Variables de sesión</title>
</head>

<body>
<p>Con estos enlaces podemos comprobar el funcionamiento de las variables de sesión.</p>
<p><a href="crea_valores.php">Crear variables de sesión</a></p>
<ul>
<li>Crearé dos variables de sesión.</li>
</ul>
<p><a href="comprueba_valores.php">Comprueba variables de sesión</a></p>
<ul>
<li>Comprobaré si las variables se han creado.
<br>Si no existen las variables nos mostrará un mensaje de advertencia.</li>
</ul>
<p><a href="destruye_valores.php">Elimina variables de sesión</a></p>
<ul>
<li>Destruiré las variables y cerraré la sesión.
<br>Si no existen las variables nos mostrará un mensaje de advertencia.</li>
</ul>
</body>
</html>
```

### crea\_valores.php

```
<?php
session_start(); // Iniciamos sesión
$_SESSION['usr']='Luis'; // Creamos variable de sesión
$_SESSION['clv']='2222'; // Creamos variable de sesión
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Crear variables</title>
</head>

<body>
<p>Se han creado las variables de sesión usr y clv. </p>
<p>Pulse <a href="inicio.php">aquí</a> para volver a la página de inicio.</p>
```

```
</body>
</html>
```

### comprueba\_valores.php

```
<?php
session_start(); // Iniciamos sesión
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Comprueba variables</title>
</head>

<body>
<?php
if (isset($_SESSION['usr'])) {
    echo '<p>Variables de sesión:<br>
        <ul>
            <li>usr -> ' . $_SESSION['usr'] . '</li>
            <li>clv -> ' . $_SESSION['clv'] . '</li>
        </ul>
        <p>';
} else {
    echo '<p>No se han encontrado las variables de sesión esperadas.</p>';
}
?>
<p>Pulse <a href="inicio.php">aquí</a> para volver a la página de inicio.</p>
</body>
</html>
```

### destruye\_valores.php

```
<?php
// Función para destruir la sesión actual
function CerrarSesion() {
    // Destruimos la cookie de sesión si existe
    if(isset($_COOKIE[session_name()])) {
        setcookie(session_name(), '', -4200, '/');
    }
    session_unset(); // Destruimos las variables de sesión
    session_destroy(); // Destruimos finalmente la información de la sesión
}

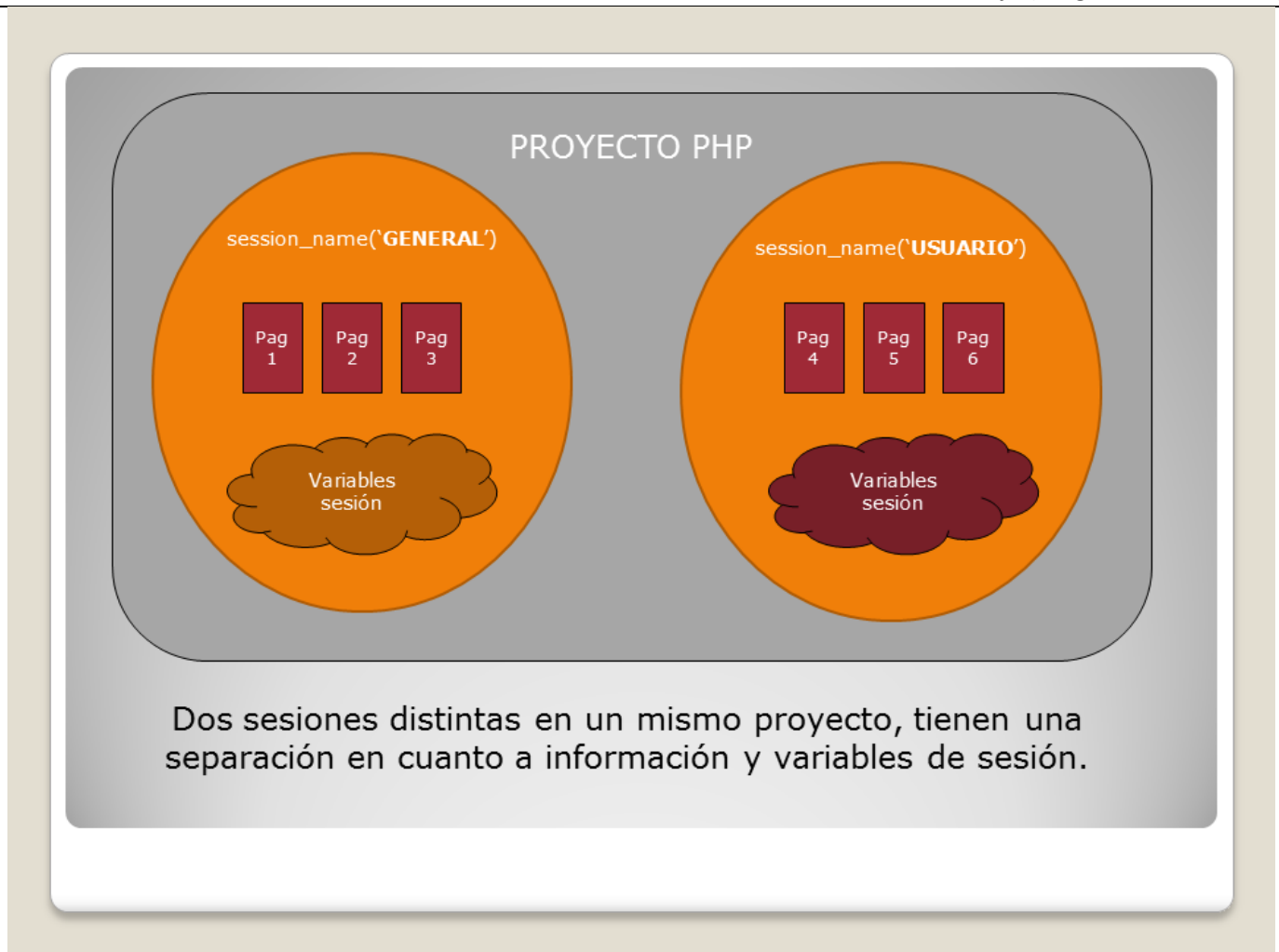
session_start(); // Iniciamos sesión

// Si se han creado variables de sesión, la cerramos y las eliminamos
if(count($_SESSION)>0) {
    cierra_sesion(); // Llamamos a la función de cierre de sesión
    $cerrada=true; // Activamos indicador de cierre
}
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Destruir variables</title>
</head>

<body>
<?php
if(isset($cerrada)) {
    echo '<p>Se ha cerrado la sesión y se han destruido los datos.</p>';
} else {
    echo '<p>Todavía no se han creado las variables de sesión.</p>';
}
?>
<p>Pulse <a href="inicio.php">aquí</a> para volver a la página de inicio.</p>
</body>
</html>
```

## Crear distintas sesiones en el proyecto

Aunque normalmente se trabaja con la misma sesión durante todo el proyecto en la que se comparten los mismos datos entre todas las páginas, podemos crear sesiones distintas para determinadas partes del mismo, lo que nos permite almacenar y compartir información distinta en cada caso.



Toda sesión se identifica por un nombre que si no se especifica se le suele asignar 'PHPSESSID'. Si antes de iniciar la sesión especificamos un nombre distinto, estaremos iniciando una sesión distinta a la anterior, con unas variables de sesión también distintas a la sesión anterior.

Este método de trabajo implica que unas páginas utilizaran un nombre de sesión y otras páginas utilizarán otro nombre de sesión distinto.

La función `session_name()` nos permite crear nombres de sesión. Devuelve o establece el nombre de sesión actual.

```
string session_name([string $name])
```

El nombre de sesión se tendrá que asignar antes de realizar la llamada a la función `session_start()`.

Veamos un ejemplo en el que según el nombre de sesión que indiquemos se accederá a unas variables de sesión distintas.

- *inicio.php*: muestra los enlaces a las páginas que crean las sesiones y a la que consulta las variables de cada sesión.
- *sesion\_meses.php*: inicia una sesión de nombre MESES y crea dos variables.
- *sesion\_fiestas.php*: inicia una sesión de nombre FIESTAS y crea dos variables.
- *consulta\_valores.php*: consulta las variables de la sesión que se le indique enviándole el nombre por medio del método GET.

## inicio.php

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Nombres de sesión</title>
</head>

<body>
<p> Con estos enlaces podemos iniciar sesiones distintas por medio del nombre de sesión.</p>
<p><a href="sesion_meses.php">Sesión MESES</a></p>
<ul>
<li>Inicia sesión de nombre 'MESES' y crea variables de sesión.</li>
</ul>
<p><a href="consulta_valores.php?s=MESES">Consultar sesión MESES</a></p>
<ul>
<li>Consulta las variables de la sesión MESES.</li>
</ul>
<HR />
<p><a href="sesion_fiestas.php">Sesión FIESTAS</a></p>
<ul>
<li>Inicia sesión de nombre 'FIESTAS' y crea variables de sesión.</li>
</ul>
<p><a href="consulta_valores.php?s=FIESTAS">Consultar sesión FIESTAS</a></p>
<ul>
<li>Consulta las variables de la sesión FIESTAS.</li>
</ul>
</body>
</html>
```

## sesion\_meses.php

```
<?php
session_name('MESES');           // Indicamos nombre de sesión
session_start();                 // Iniciamos sesión
// Creamos variables
$_SESSION['enero']=31;
$_SESSION['febrero']=28;
$_SESSION['marzo']=31;
$_SESSION['abril']=30;
$_SESSION['mayo']=30;
?>

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Crear variables sesión MESES</title>
</head>

<body>
<p>Se han creado 5 variables en la sesión MESES.</p>
<p>Pulse <a href="inicio.php">aquí</a> para volver a la página de inicio.</p>
</body>
</html>
```

## sesion\_fiestas.php

```
<?php
session_name('FIESTAS');         // Indicamos nombre de sesión
session_start();                 // Iniciamos sesión
// Creamos variables
$_SESSION['Valencia']='Las Fallas';
$_SESSION['Castellon']='La Magdalena';
$_SESSION['Alicante']='Las Hogueras';
$_SESSION['Murcia']='Semana Santa';
$_SESSION['Almeria']='La Feria';
?>

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Crear variables sesión FIESTAS</title>
</head>

<body>
<p>Se han creado 5 variables en la sesión FIESTAS.</p>
<p>Pulse <a href="inicio.php">aquí</a> para volver a la página de inicio.</p>
</body>
</html>
```

## consulta\_valores.php

```

<?php
session_name($_GET['s']);          // Indicamos nombre de sesión recibido por el método GET
session_start();                  // Iniciamos sesión
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title> Consultar variables sesión <?php echo $_GET['s'];?></title>
</head>

<body>
<p>Variables creadas en la sesión <?php echo $_GET['s'];?></p>
<ul>
<?php
foreach($_SESSION as $clave => $valor) {
    echo '<li>','<li>', $clave, '->', $valor, '</li>';
}
?>
</ul>
<p>Pulse <a href="inicio.php">aquí</a> para volver a la página de inicio.</p>
</body>
</html>

```

El resultado sería:

Variables creadas en la sesión MESES	Variables creadas en la sesión FIESTAS
<ul style="list-style-type: none"> <li>• enero-&gt;31</li> <li>• febrero-&gt;28</li> <li>• marzo-&gt;31</li> <li>• abril-&gt;30</li> <li>• mayo-&gt;30</li> </ul>	<ul style="list-style-type: none"> <li>• Valencia-&gt;Las Fallas</li> <li>• Castellon-&gt;La Magdalena</li> <li>• Alicante-&gt;Las Hogueras</li> <li>• Murcia-&gt;Semana Santa</li> <li>• Almeria-&gt;La Feria</li> </ul>

## Trabajar con varias sesiones en el mismo script

Si en algún momento dado por la causa que sea, necesitamos cambiar la sesión actual por una nueva o necesitamos recuperar la información de otra sesión iniciada, tendremos que guardar y cerrar los datos de la sesión activa y a continuación iniciar la nueva sesión.

Para poder realizar estos cambios necesitamos conocer los identificadores de las sesiones con los que deseamos trabajar y un método que nos permita cerrar la sesión actual sin perder los datos. Estas operaciones las llevaremos a cabo con las funciones: `session_id()` y `session_write_close()`.

### `session_id()`

Toda sesión iniciada tiene un nombre asociado que la identifica durante todo el trabajo con sus variables de sesión.

Este identificador lo podemos variar para trabajar con varias sesiones distintas durante la misma ejecución, lo que nos permite mantener una separación de datos.

```
string session_id([identificador])
```

Esta función nos sirve para establecer u obtener el valor de la id de la sesión para la sesión actual.

### `session_write_close()`

Si lo que pretendemos es cambiar a una sesión distinta de la actual, primero tendremos que guardar los valores de la sesión actual y luego iniciar la nueva sesión indicando su id.

```
void session_write_close()
```

Esta función, guarda la información de la sesión actual y luego la cierra.

Por lo tanto si deseamos trabajar con multisesiones, tendremos que utilizar una combinación de estas funciones, junto con session\_start().

```
<?php
$session_1="Primera";
$session_2="Segunda";

session_id($session_1);           // Establecemos el id de la primera sesión
session_start();                 // Iniciamos sesión
$_SESSION['nombre']='Luis';       // Creamos variable de sesión
// Mostramos el identificador de la sesión y el contenido de la variable
echo 'Sesión(' .session_id().')  nombre: ' .$_SESSION['nombre'].'<br>';
session_write_close();           // Guardamos los datos de la sesión y la cerramos
session_id($session_2);         // Establecemos el id de la segunda sesión
session_start();                 // Iniciamos sesión
$_SESSION['nombre']='Ana';       // Creamos variable de sesión
// Mostramos el identificador de la sesión y el contenido de la variable
echo 'Sesión(' .session_id().')  nombre: ' .$_SESSION['nombre'].'<br>';
session_write_close();           // Guardamos los datos de la sesión y la cerramos
session_id($session_1);         // Establecemos el id de la primera sesión
session_start();                 // Iniciamos sesión
// Mostramos el identificador de la sesión y el contenido de la variable
echo 'Sesión(' .session_id().')  nombre: ' .$_SESSION['nombre'].'<br>';
?>
```

El resultado sería:

```
Sesión(Primera) nombre: Luis
Sesión(Segunda) nombre: Ana
Sesión(Primera) nombre: Luis
```