

Programación III

TUW – TUR

- 2022 -

PHP: DATOS PERSISTENTES

Cookies y Sesiones

Cookies

Un cookie es un archivo creado por un sitio de Internet para almacenar información en su equipo, por ejemplo, sus preferencias cuando visita el sitio.

Por Ejemplo

- Si consulta la programación de un vuelo en el sitio Web de una línea aérea, el sitio podría crear un cookie que tuviera su itinerario.
- Los Cookies también pueden almacenar información que le identifique personalmente. La información que puede identificarle personalmente se puede usar para identificarle o ponerse en contacto con usted, por ejemplo, su nombre, su dirección de correo electrónico o de su domicilio o trabajo, o su número de teléfono entran dentro de este tipo de información
- etc.

Consideraciones

- Sin embargo, un sitio Web sólo tiene acceso a la información personal que usted le proporcione.
 - Por ejemplo, un sitio Web no puede determinar su nombre de correo electrónico a menos que usted se lo indique. Tampoco puede obtener acceso a otra información de su equipo.

Tipos de Cookies

- Cookies Persistentes
- Cookies Temporales
- Cookies de Origen
- Cookies de Terceros

Cookie persistente

Un cookie persistente es el que se almacena como archivo en un equipo y permanece allí cuando cierra el browser.

El cookie puede ser leído por el sitio Web que lo creó cuando vuelva a visitarlo.

Cookie temporal

Un cookie temporal o de sesión se almacena únicamente durante la sesión de exploración actual y se elimina del equipo cuando se cierra el navegador.

Cookie de origen

Un cookie de origen se origina en el sitio Web que está viendo en ese momento o se envía a dicho sitio. Estos cookies se suelen usar para almacenar información, por ejemplo, sus preferencias al visitar el sitio

Cookie de terceros

Un cookie de terceros se origina en un sitio Web diferente al que está viendo en ese momento o se envía a dicho sitio.

Un sitio Web tercero normalmente proporciona cierto contenido al sitio Web que está viendo.

Opciones de Trabajo

La mayoría de los Browsers permite el uso de cookies; sin embargo, puede cambiar su configuración de privacidad para especificar que le pregunte antes de colocar un cookie en su equipo (de este modo, puede autorizarlo o bloquearlo); o puede impedir que el navegador acepte los cookies.

IMPORTANTE

Una vez que un cookie se guarda en el equipo, sólo el sitio Web que lo creó puede leerlo.

Cookies en PHP

```
int setcookie (string nombre [, string valor  
                [, int fin [, string camino  
                [, string dominio  
                [, int seguro]]]])
```

```
setcookie("PruebaCookie",  
          "expiraré dentro de una hora",  
          time() + 3600);
```

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

Parámetros (1)

`setCookie (nombre, [valor [, fecha expiración, [, ruta [, dominio [. Seguridad]]]]])`

Nombre: Es el nombre que se le asigna a la cookie. Es el único valor obligatorio.

Valor: Son los datos que almacenamos en el navegador visitante.

Fecha de expiración: Es el número de segundos que han de pasar antes de que la cookie se borre automáticamente. Por ejemplo, si queremos que una cookie expire en 2 horas, (7200 segundos), tendríamos que poner la fecha de expiración como `time() +7200`.

Parámetros (2)

Ruta: Este parámetro nos servirá en el caso de que queramos restringir el acceso de la cookie a una determinada ruta de nuestro servidor. (Esta función tan solo nos será útil en el caso de que estemos compartiendo el servidor con otros usuarios y cada uno tenga sus propias carpetas).

Dominio: Es como el anterior (ruta), pero en vez de restringir o dar permisos a carpetas lo hace con dominios, este parámetro por defecto sólo enviara la cookie al dominio original. Si queremos utilizar esta cookie en otro dominios deberemos indicarlo en este campo.

Seguridad: Este parámetro es el que determina la seguridad de la cookie, tiene 2 valores, 0 y 1. Con el valor 1 indica que la cookie sólo se enviará vía servidor seguro (SSL) y con el 0, indicará que podrá ser mandada con cualquier conexión.

Cookie Habilidades?

```
<?php
if (!isset ($_GET['testcookie'])) {
    setcookie ("testcookie", "test value");
    header ("Location: {$_SERVER['PHP_SELF']}?testcookie=1");
    exit;
} elseif (isset ($_COOKIE['testcookie'])) {
    setcookie ("testcookie");
    echo "Ud. tiene las cookies habilitadas";
} else {
    echo "Ud. no soporta cookies.";
}
?>
```



Sesiones

Las sesiones nos permiten registrar un número arbitrario de variables que se conservan durante toda la visita de un usuario a un sitio web.

Dichas variables pueden ser diferentes para cada usuario, ya que están referenciadas por un identificador único que se le asigna a cada visitante.

Sesiones

- Una sesión es una cantidad de información que se almacena dentro del servidor con un identificador único de usuario.
- Este id es enviado automáticamente al usuario por medio de cookies.

Sesiones

- Dentro de ésta sesión se pueden almacenar los valores de las variables y disponer de ellos en cualquiera de las páginas en las que se inicie la sesión.
- Una ventaja extra es que al cambiar el valor de las variables, automáticamente cambia su valor para toda la sesión.

Sesiones

- En otras palabras, una sesión es una manera de almacenar variables de manera temporal, semejante a una cookie, pero con ciertas diferencias:
 - Las COOKIES se almacenan en la PC del usuario y pueden desactivarse.
 - Las sesiones se almacenan temporalmente en el servidor, en un archivo que se crea en el momento en que almacenamos la variable.

Sesiones

```
bool session_start(void);  
bool session_register (mixed name [, mixed ...])  
bool session_destroy(void);  
session_id()  
session_unregister('variable')
```

Ejemplo:

Carrito de Compras

Introducimos el manejo de sesiones a través de un ejemplo en el que se verá cómo trabaja un *carrito de compras*.

Scripts PHP Definidos

- **Catalogo.php:** Este script muestra los artículos para la venta. Mostrando los productos agregados y no agregados. Además permite el agregado y eliminación de un artículo al carrito.
- **Agregacar.php:** A través de este script se agrega un nuevo artículo al carrito
- **Vercarrito.php:** Muestra los elementos agregados al carrito y permite modificar, de cada artículo, la cantidad a comprar.
- **Borracar.php:** Elimina un artículo del carrito.

Esquema BD

```
CREATE TABLE `catalogo` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `producto` VARCHAR( 100 ) NULL ,  
  `precio` DECIMAL NULL  
) ENGINE = INNODB;
```

Catalogo.php

```
<?php
```

```
session_start();
```

```
//conectamos a la base de datos
```

```
$conexion = mysqli_connect("localhost", "usuario", "password", "db");
```

```
// rescatamos los valores guardados en la variable de sesión
```

```
// (si es que hay alguno, cosa que comprobamos con isset)
```

```
// y los asignamos a $carro.
```

```
// Si no existen valores, ponemos a false el valor de $carro
```

```
if (isset($_SESSION['carro'])) $carro=$_SESSION['carro'];
```

```
    else $carro=false;
```

```
//y hacemos la consulta
```

```
$qry=mysqli_query($conexion, "select * from catalogo order by producto asc");
```

```
?>
```


Catalogo.php

```
<html>
<head>
<title>CAT&Aacute;LOGO</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
.catalogo {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 9px;
    color: #333333;
}
-->
</style>
</head>
```

Catalogo.php

```
<body>
```

```
<table width="272" align="center" cellpadding="0" cellspacing="0" style="border: 1px solid #000000;">
```

```
<tr valign="middle" bordercolor="#FFFFFF" bgcolor="#DFDFDF" class="catalogo">
```

```
<td width="170"><strong>Producto</strong></td>
```

```
<td width="77"><strong>Precio</strong></td>
```

```
<td width="25" align="right">
```

```
<a href="vercarrito.php?<?php echo session_id()?>" title="Ver el contenido del carrito"></a>
```

```
</td>
```

```
</tr>
```

Catalogo.php

```
<?php
```

```
// mostramos todos nuestros artículos, viendo si han sido agregados o no a  
// nuestro carro de compra
```

```
while ($row=mysqli_fetch_assoc($qry)){
```

```
?>
```

```
<tr valign="middle" class="catalogo">
```

```
<td><?php echo $row['producto'] ?></td>
```

```
<td><?php echo $row['precio'] ?></td>
```

```
<td align="center">
```

```
<?php
```

```
if (!$carro || !isset($carro[md5($row['id'])]['identificador']) ||  
$carro[md5($row['id'])]['identificador']!=md5($row['id'])){
```

```
// si el producto no ha sido agregado, mostramos la imagen de no agregado,  
// linkeada a nuestra página de agregar producto y transmitiéndole a  
// dicha página el id del artículo y el identificador de la sesión
```

```
?>
```

Catalogo.php

```
<a href="agregar.php?sid=<?php echo SID ?>&id=<?php echo $row['id'];  
?>">  
</a>  
<?php } else  
//en caso contrario mostramos la otra imagen linkeada, a la página que sirve  
// para borrar el artículo del carro.  
{?>  
<a href="borracar.php?sid=<?php echo SID ?>&id=<?php echo $row['id'];  
?>">  
</a>  
<?php } ?>  
</td>  
</tr>  
<?php } ?>  
</table>  
</body>  
</html>
```

Agregacar.php

```
<?php
```

```
session_start();
```

```
// con session_start() creamos la sesión si no existe o la retomamos si ya ha  
// sido creada
```

```
extract($_REQUEST);
```

```
// la función extract toma las claves de una matriz asociativa y las convierte  
// en nombres de variable, asignándoles a esas variables valores iguales a  
// los que tenía asociados en la matriz. Es decir, convierte a $_REQUEST['id'] en  
// $id, sin que tengamos que tomarnos el trabajo de escribir $id=$_REQUEST['id'];
```

```
$conexion = mysqli_connect("localhost", "usuario", "password", "db");
```

```
// incluimos la conexión a nuestra base de datos
```

Agregacar.php

```
if (!isset($cantidad)) { $cantidad=1; }
```

```
// Como también vamos a usar este archivo para actualizar las cantidades,  
// hacemos que cuando la misma no esté indicada sea igual a 1
```

```
$sql = "select * from catalogo where id=$id";  
$qry = mysqli_query($conexion, $sql);  
$row = mysqli_fetch_array($qry);
```

```
// Si ya hemos introducido algún producto en el carro lo tendremos guardado  
// temporalmente en el array superglobal $_SESSION['carro'], de manera  
// que rescatamos los valores de dicho array y se los asignamos a la variable  
// $carro, previa comprobación con isset de que $_SESSION['carro'] ya haya  
// sido definida.
```

Agregacar.php

```
If (isset($_SESSION['carro']))  
    $carro=$_SESSION['carro'];
```

```
// Ahora introducimos el nuevo producto en la matriz $carro, utilizando como  
// índice el id del producto en cuestión, encriptado con md5.  
// Utilizamos md5 porque genera un valor alfanumérico que luego, cuando  
// busquemos un producto en particular dentro de la matriz, no podrá ser  
// confundido con la posición que ocupa dentro de dicha matriz, como podría  
// ocurrir si fuera sólo numérico.  
// Cabe aclarar que si el producto ya había sido agregado antes, los nuevos  
// valores que le asignemos reemplazarán a los viejos.  
// Al mismo tiempo, y no porque sea estrictamente necesario sino a modo de  
// ejemplo, guardamos más de un valor en la variable $carro, valiéndonos de  
// nuevo de la herramienta array.
```

```
$carro[md5($id)] = array('identificador'=>md5($id),  
    'cantidad'=>$cantidad,'producto'=>$row['producto'],  
    'precio'=>$row['precio'],'id'=>$id);
```

Agregacar.php

```
// Ahora dentro de la sesión ($_SESSION['carro']) tenemos sólo los valores  
// que teníamos (si es que teníamos alguno) antes de ingresar a esta página y  
// en la variable $carro tenemos esos mismos valores más el que acabamos de  
// sumar. De manera que tenemos que actualizar (reemplazar) la variable de  
// sesión por la variable $carro.
```

```
$_SESSION['carro']=$carro;
```


Agregacar.php

```
// Y volvemos a nuestro catálogo de artículos. La cadena devuelta por  
// session_id() representa al identificador de la sesión, que, dependiendo de la  
// configuración del servidor y de si el usuario tiene o no activadas las cookies  
// puede no ser necesario pasarla por la url.  
// Pero para que nuestro carro funcione, independientemente de esos factores,  
// conviene escribirla siempre.
```

```
header ("Location:catalogo.php?".session_id());
```

```
?>
```

Vercarrito.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin título</title>
</head>
<body>
<?php
session_start();
// Iniciamos o retomamos la sesión
if (isset($_SESSION['carro'])) $carro=$_SESSION['carro'];
    else $carro=false;

// La asignamos a la variable $carro si existe o ponemos a false $carro en caso
// contrario

?>
```

Vercarrito.php

```
<html> <head>
<title>PRODUCTOS AGREGADOS AL CARRITO</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
.tit {
    font-family: Verdana, Arial, Helvetica, sans-serif;  font-size: 9px;
    color: #FFFFFFF;
}
.prod {
    font-family: Verdana, Arial, Helvetica, sans-serif;  font-size: 9px;
    color: #333333;
}
h1 {
    font-family: Verdana, Arial, Helvetica, sans-serif;  font-size: 20px;
    color: #990000;
}
-->
</style>
</head>
```

Vercarrito.php

```
<body>
<h1 align="center">Carrito</h1>
<?php
If ($carro){
// si el carro no está vacío, mostramos los productos
?>
<table width="720" border="0" cellspacing="0" cellpadding="0" align="center">
<tr bgcolor="#333333" class="tit">
<td width="105">Producto</td>
<td width="207">Precio</td>
<td colspan="2" align="center">Cantidad de Unidades</td>
<td width="100" align="center">Borrar</td>
<td width="159" align="center">Actualizar</td>
</tr>
<?php
$color = array("#ffffff", "#F0F0F0");
$contador = 0;

// Las 2 líneas anteriores sirven sólo para hacer una tabla con colores alternos.
```

Vercarrito.php

```
// Antes de recorrer todos los valores de la matriz $carro, ponemos a cero la  
// variable $suma, en la que iremos sumando los subtotales del costo de cada  
// ítem por la cantidad de unidades que se especifiquen.
```

```
foreach ($carro as $k => $v){
```

```
// Recorremos la matriz que tiene todos los valores del carro, calculamos el  
// subtotal y el total
```

```
$subto = $v['cantidad']*$v['precio'];  
$suma = $suma+$subto;  
$contador++;
```

```
// este es el contador que usamos para los colores alternos
```

```
?>
```

Vercarrito.php

```
<form name="a<?php echo $v['identificador'] ?>" method="post"
action="agregar.php?sid=<?php echo session_id() ?>" id="a<?php echo
$v['identificador'] ?>">
<tr bgcolor="<?php echo $color[$contador]; ?>" class='prod'>
<td><?php echo $v['producto'] ?></td> <td><?php echo $v['precio'] ?></td>
<td width="43" align="center"><?php echo $v['cantidad'] ?></td>
<td width="136" align="center">
<input name="cantidad" type="text" id="cantidad" value="<?php echo $v['cantidad']
?>" size="8">
<input name="id" type="hidden" id="id" value="<?php echo $v['id'] ?>"> </td>
<td align="center">
<a href="borrar.php?<?php echo session_id() ?>&id=<?php echo $v['id'] ?>">
</a></td>
<td align="center">
<input name="imageField" type="image" src="actualizar.gif" width="20" height="20"
border="0"></td>
</tr></form>
<?php
// por cada item creamos un formulario que submite a agregar producto y un link
// que permite eliminarlos
}?>
</table>
```

Vercarrito.php

```
<div align="center"><span class="prod">Total de Artículos: <?php echo count($carro);  
// El total de ítems va a ser igual a la cantidad de elementos que tenga la matriz  
// $carro, valor que obtenemos con la función count o con sizeof  
<?>  
</span></div><br>  
<div align="center"><span class="prod">Total: $<?php echo number_format($suma,2);  
// Mostramos el total de la variable $suma formateándola a 2 decimales  
<?>  
</span></div><br>  
<div align="center"><span class="prod">Continuar la selección de productos</span>  
<a href="catalogo.php?<?php echo SID;?>">  
</a>  
</div>
```

Vercarrito.php

```
<?php }else{ ?>
```

```
<p align="center"> <span class="prod">No hay productos seleccionados</span>
```

```
<a href="catalogo.php?<?php echo session_id();?>">
```

```
</a>
```

```
</p>
```

```
<?php }?>
```

```
</body>
```

```
</html>
```


Borrar.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin título</title>
</head>
<body>
<?php
session_start();
//Con session_start() creamos la sesión si no existe o la retomamos si ya ha sido
//creada

extract($_GET);
// Como antes, usamos extract() por comodidad, pero podemos no hacerlo
// tranquilamente
```

Borrarcar.php

```
$carro = $_SESSION['carro'];
```

```
// Asignamos a la variable $carro los valores guardados en la sesión
```

```
unset($carro[md5($id)]);
```

```
// la función unset borra el elemento de un array que le
```

```
// pasemos por parámetro. En este caso la usamos para borrar el elemento cuyo
```

```
// id le pasemos a la página por la url.
```

```
$_SESSION['carro']=$carro;
```

```
// Finalmente, actualizamos la sesión, como hicimos cuando agregamos un
```

```
// producto y volvemos al catálogo
```

```
header("Location:catalogo.php?".session_id());
```

```
?>
```

```
</body>
```

```
</html>
```

Esquema básico de autenticación en PHP usando Sesiones

- Normalmente cuando autenticamos en PHP lo que pretendemos es controlar qué usuarios acceden y qué usuarios no acceden a una aplicación, Para eso se siguen estos pasos.
- Petición de nombre de usuario y contraseña.
- Comprobación de los datos anteriores.
- En caso correcto, se establece la sesión.
- En caso incorrecto se vuelve a solicitar usuario y contraseña.

