

# PROGRAMACIÓN II



**SEMANA 06**  
**SESIONES**

Eric Gustavo Coronel Castillo  
[ecoronel@uch.edu.pe](mailto:ecoronel@uch.edu.pe)  
[gcoronelc.blogspot.com](http://gcoronelc.blogspot.com)

# SESIONES

Una de las funcionalidades fundamentales que tenemos en PHP es el manejo de sesiones, esto nos permite tener datos persistentes durante todo el tiempo que el usuario permanece conectado al servidor Web.

## Índice

<b>1</b>	<b>¿QUÉ SON LAS SESIONES?</b> .....	<b>4</b>
<b>2</b>	<b>MANEJO DE SESIONES</b> .....	<b>4</b>
2.1	FUNCIÓN: <code>SESION_START()</code> .....	4
2.2	FUNCIÓN: <code>SESSION_ID()</code> .....	6
2.3	ARREGLO: <code>\$_SESSION</code> .....	7
2.4	FUNCIÓN: <code>SESSION_UNSET()</code> .....	9
2.5	FUNCIÓN: <code>SESSION_DESTROY()</code> .....	9
<b>3</b>	<b>EJEMPLO APLICATIVO</b> .....	<b>13</b>

## 1 ¿Qué son las sesiones?

---

Las sesiones son un mecanismo basado en cookies que permiten identificar a los usuarios que acceden a un sitio Web, de esta manera almacenar información referente a sus transacciones.

Un ejemplo típico donde se aplican sesiones, son las aplicaciones de comercio electrónico, en estas aplicaciones se debe llevar un registro de los productos que ha agregado a su canasta de compras.

## 2 Manejo de Sesiones

---

### 2.1 Función: sesión\_start()

#### Sintaxis

```
boolean session_start ( void )
```

#### Valor de Retorno

Siempre devuelve TRUE.

Esta función en caso de que el visitante no tenga una sesión la crea y si ya tiene una sesión continua en ella.

La propagación de una sesión es por cookies, claro que de eso se encarga la función, por ello es necesario llamar a esta función antes de enviar cualquier cabecera HTTP, esto es antes de comenzar a imprimir nuestra página, de lo contrario habrá un error al tratar de crear una cookie.

Una vez iniciada la sesión podremos almacenar información de la sesión en el arreglo **\$\_SESSION**. Este es un arreglo asociativo, además es super-global lo que quiere decir que su alcance se extiende a todo el ámbito de la aplicación y no tendrás que declararlo como global al utilizarlo dentro de funciones.

## Ejemplo 1

En este ejemplo se ilustra la manera **como no se debe usar** la función **session\_start()**.

Archivo: ejm01.php

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Ejemplo de Sesión</title>
</head>
<body bgcolor="#D2EBF7" style="font-family: Tahoma;">
  <h3>Ejemplo de Sesión</h3>
  <?php
    session_start();
    echo "Estoy en una sesión";
  ?>
</body>
</html>
```

El resultado que se obtiene es:

### Ejemplo de Sesión

**Warning:** session\_start() [[function.session-start](#)]: Cannot send session cookie - headers already sent by (output started at C:\PHP100\cap08\ejm01.php:8) in C:\PHP100\cap08\ejm01.php on line 9

**Warning:** session\_start() [[function.session-start](#)]: Cannot send session cache limiter - headers already sent (output started at C:\PHP100\cap08\ejm01.php:8) in C:\PHP100\cap08\ejm01.php on line 9  
Estoy en una sesión

## 2.2 Función: session\_id()

### Sintaxis

```
string session_id ( [string id])
```

### Valor de Retorno

Retorna un cadena con el id de sesión.

Esta función se utiliza para leer o cambiar el id de la sesión actual.

### Ejemplo 2

Este ejemplo ilustra la manera como se debe trabajar con sesiones.

Archivo: ejm02.php

```
<?php
    session_start();
    $idSesion = session_id();
?>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Ejemplo de Sesión</title>
    </head>
    <body bgcolor="#D2EBF7" style="font-family: Tahoma;">
        <h3>Ejemplo de Sesión</h3>
        <p>Estoy en una sesión</p>
        <p>El Id de sesión es: <?php echo($idSesion) ?></p>
    </body>
</html>
```

El resultado se ilustra a continuación:

### **Ejemplo de Sesión**

Estoy en una sesión

El Id de sesión es: 62ac3c13c1b8d7184c27f6bf04852499

## 2.3 Arreglo: \$\_SESSION

Es una matriz asociativa que contiene las variables de sesión disponibles en la aplicación actual.

Se trata de una variable super-global o global automática. Esto simplemente quiere decir que está disponible en todos los contextos a lo largo de una aplicación.

### Ejemplo 3

En el siguiente ejemplo se ilustra el uso del arreglo \$\_SESSION.

El usuario ingresa nombres de artículo y sus respectivos precios, cada nombre de artículo se utiliza como clave en el arreglo asociativo \$\_SESSION.

Archivo: ejm03.php

```
<?php
session_start();

if( isset( $_POST["enviar"] ) ) {
    $_SESSION[ $_POST["articulo"] ] = $_POST["precio"];
}
if( isset( $_POST["limpiar"] ) ) {
    session_unset();
    session_destroy();
}
$pagina = "http://" . $_SERVER['HTTP_HOST'] . $_SERVER['SCRIPT_NAME'];
?>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Variables de Sesión</title>
  </head>
  <body bgcolor="#D2EBF7" style="font-family: Tahoma;">
    <h3>Datos de Artículo</h3>
    <form method="post" action="<?php echo($pagina) ?>">
      <table width="240">
        <tr>
          <td width="76">Artículo</td>
          <td width="152"><input type="text" name="articulo"></td>
        </tr>
        <tr>
          <td>Precio</td>
          <td><input type="text" name="precio"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```

</table>
<p>
  <input name="enviar" type="submit" id="enviar" value="Enviar">
  <input type="submit" name="limpiar" id="limpiar" value="Limpiar">
</p>
</form>
<?php if( count($_SESSION) > 0 ) { ?>
<table border='1' width='200'>
  <tr>
    <th>Artículo</th>
    <th>Precio</th>
  </tr>
  <?php foreach ( $_SESSION as $clave => $valor ) { ?>
  <tr>
    <td><?php echo($clave) ?></td>
    <td><?php echo($valor) ?></td>
  </tr>
  <?php } ?>
</table>
<?php } ?>
</body>
</html>

```

El resultado se ilustra a continuación:

### Datos de Artículo

Artículo

Precio

Artículo	Precio
Televisor	580
Impresora	278
Disco Duro	340

## 2.4 Función: `session_unset()`

### Sintaxis

```
void session_unset ( void )
```

### Valor de Retorno

No retorna ningún valor.

Esta función elimina y libera el espacio ocupado por todas las variables de sesión actualmente registradas.

## 2.5 Función: `session_destroy()`

### Sintaxis

```
bool session_destroy ( void )
```

### Valor de Retorno

Retorna un valor lógico que indica el estado de su ejecución.

Esta función destruye todos los datos asociados con la sesión actual. No destruye ninguna de las variables globales asociadas a la sesión ni la cookie.

Retorna TRUE si se ha destruido la sesión correctamente y FALSE en caso de haber algún problema.

### Ejemplo 4

En este ejemplo se utilizará dos programas, los cuales se detallan a continuación:

Programa	Descripción
<b>ejm04.php</b>	Este programa es prácticamente igual al programa <b>ejm03.php</b> , salvo por el link hacia el programa <b>ejm04_delete.php</b> .
<b>ejm04_delete.php</b>	Este programa se encarga de eliminar la sesión actual, y luego redirecciona la ejecución al programa <b>ejm04.php</b> .



```
<?php
session_start();

if( isset( $_POST["enviar"] ) ) {
    $_SESSION[ $_POST["articulo"] ] = $_POST["precio"];
}
$pagina = "http://" . $_SERVER['HTTP_HOST'] . $_SERVER['SCRIPT_NAME'];
?>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Variables de Sesión</title>
</head>
<body bgcolor="#D2EBF7" style="font-family: Tahoma;">
    <h3>Datos de Artículo</h3>
    <form method="post" action="<?php echo($pagina) ?>">
        <table width="240">
            <tr>
                <td width="76">Artículo</td>
                <td width="152"><input type="text" name="articulo"></td>
            </tr>
            <tr>
                <td>Precio</td>
                <td><input type="text" name="precio"></td>
            </tr>
        </table>
        <p>
            <input name="enviar" type="submit" id="enviar" value="Enviar">
        </p>
    </form>
    <?php if( count($_SESSION) > 0 ) { ?>
    <table border='1' width='200'>
        <tr>
            <th>Artículo</th>
            <th>Precio</th>
        </tr>
        <?php foreach ( $_SESSION as $clave => $valor ) { ?>
        <tr>
            <td><?php echo($clave) ?></td>
            <td><?php echo($valor) ?></td>
        </tr>
        <?php } ?>
    </table>
```

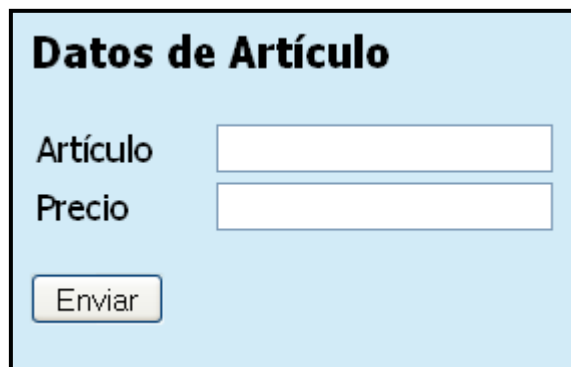
```
<a href="ejm04_delete.php">Eliminar Sesión</a>
<?php } ?>
</body>
</html>
```

Archivo: ejm04\_delete.php

---

```
<?php
session_start();
session_unset();
session_destroy();
header("location: ejm04.php");
?>
```

Cuando ejecuta el programa **ejm04.php**, obtiene la siguiente interfaz:



The screenshot shows a web form titled "Datos de Artículo" in a light blue box. It contains two input fields: "Artículo" and "Precio". Below these fields is a button labeled "Enviar".

Luego de ingresar algunos datos obtenemos la siguiente interfaz:

### Datos de Artículo

Artículo

Precio

Enviar

Artículo	Precio
Coca Cola	3
Combo Familiar	35

[Eliminar Sesión](#)

El enlace **Eliminar Sesión** ejecuta el programa **ejm04\_delete.php** que se encarga de eliminar la sesión y todas sus variables, luego redirecciona el control al programa **ejm04.php**, y nuevamente aparece el formulario, pero ya no los artículos, porque han sido eliminados todos los datos de la sesión.

### 3 Ejemplo Aplicativo

---

En este ejemplo se ilustra la implementación de un carrito de compras básico, esta demostración está compuesta por los siguientes programas:

Programa	Descripción
<b>inicio.html</b>	Este programa muestra el formulario para que el usuario ingrese sus datos de inicio de sesión, y los envía al programa <b>inicio.php</b> .
<b>inicio.php</b>	Este programa procesa los datos enviados por el programa <b>inicio.html</b> , específicamente inicia la sesión y crea las variables de sesión respectiva.
<b>carrito.php</b>	Este programa muestra el contenido del carrito, y enlaces hacia los programas <b>agregararticulo.php</b> y <b>cerrarsesion.php</b> .
<b>agregaarticulo.php</b>	Este programa muestra un formulario para agregar un nuevo artículo al carrito de compras, luego redirecciona el control al programa <b>carrito.php</b> .
<b>cerrarsesion.php</b>	Este programa cierra la sesión y luego redirecciona el control al programa <b>inicio.html</b> .

Archivo: inicio.html

---

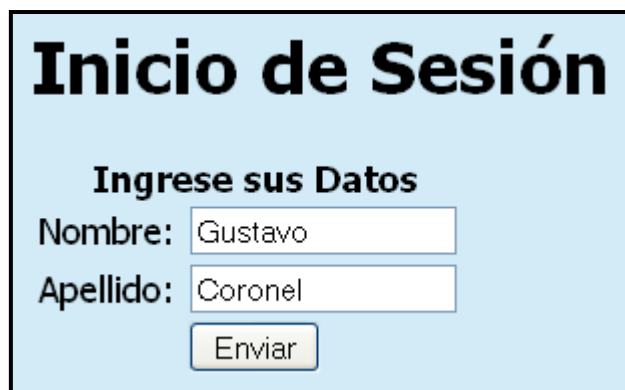
```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Inicio de Sesión</title>
  </head>
  <body bgcolor="#D2EBF7" style="font-family: Tahoma;">
    <h1>Inicio de Sesión</h1>
    <form method="post" action="inicio.php" >
      <table width="200">
        <caption><strong>Ingresa tus Datos</strong></caption>
        <tr>
          <td>Nombre:</td>
          <td>
            <input type="text" name="nombre" size="15" maxlength="15">
          </td>
        </tr>
        <tr>
          <td>Apellido:</td>
```

```

        <td>
            <input type="text" name="apellido" size="15" maxlength="15">
        </td>
    </tr>
    <tr>
        <td colspan="2" align="center">
            <input name="enviar" type="submit" value="Enviar" id="enviar">
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Este programa permite al usuario identificarse, presenta la siguiente interfaz para que ingrese su nombre y apellido:



Archivo: inicio.php

```

<?php
session_start();
if( ! isset( $_POST["nombre"] ) || ! isset( $_POST["apellido"] ) ) {
    $destino = "inicio.html";
} else {
    $_SESSION["nombre"] = $_POST["nombre"];
    $_SESSION["apellido"] = $_POST["apellido"];
    $_SESSION["carrito"] = array();
    $destino = "carrito.php";
}
header( "location: $destino" );
?>

```

Este programa recibe los datos ingresados en el formulario del programa **inicio.html**, inicializa la sesión y luego redirecciona el control hacia el programa **carrito.php**.

Archivo: carrito.php

```
<?php
session_start();
if( ! isset($_SESSION["nombre"]) ){
    header( "location: inicio.html" );
    return;
}
$carritoVacio = (count($_SESSION["carrito"]) == 0);
$nombre = "{$_SESSION["nombre"]} {$_SESSION["apellido"]}";
?>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Inicio de Sesión</title>
</head>
<body bgcolor="#D2EBF7" style="font-family: Tahoma;">
    <h2>Usuario: <?php echo( $nombre ) ?></h2>
    <table width="358">
        <tr>
            <td width="170" align="center">
                <a href="agregararticulo.php">Agregar Artículo</a>
            </td>
            <td width="176" align="center">
                <a href="cerrarsesion.php">Cerrar Sesión</a>
            </td>
        </tr>
    </table>
    <?php if( $carritoVacio ) { ?>
    <h3>Carrito Vacío</h3>
    <?php } else { ?>
    <table width='447' border='1'>
        <tr>
            <th width="188">Artículo</th>
            <th width="73">Precio</th>
            <th width="65">Cant.</th>
            <th width="93">Subtotal</th>
        </tr>
        <?php foreach ( $_SESSION["carrito"] as $rec ) { ?>
        <tr>
            <td><?php echo($rec["articulo"]) ?></td>
            <td><?php echo($rec["precio"]) ?></td>
```

```

        <td><?php echo($rec["cantidad"]) ?></td>
        <td><?php echo($rec["subtotal"]) ?></td>
    </tr>
    <?php } ?>
</table>
<?php } ?>
</body>
</html>

```

Después de agregar varios artículos este programa tendrá el siguiente resultado:

**Usuario: Gustavo Coronel**

[Agregar Artículo](#)
[Cerrar Sesión](#)

Artículo	Precio	Cant.	Subtotal
Televisor	589	4	2356
Teclado	30	3	90
Impresora	230	6	1380

Archivo: agregararticulo.php

```

<?php
session_start();
if( ! isset($_SESSION["nombre"]) ){
    header( "location: inicio.html" );
    return;
}
if( isset( $_POST["enviar"] ) ) {
    $articulo = $_POST["articulo"];
    $rec["articulo"] = $_POST["articulo"];
    $rec["precio"] = $_POST["precio"];
    $rec["cantidad"] = $_POST["cantidad"];
    $rec["subtotal"] = $rec["precio"] * $rec["cantidad"];
    $_SESSION["carrito"][$articulo] = $rec;
    header( "location: carrito.php" );
    return;
}
$nombre = "{$_SESSION["nombre"]} {$_SESSION["apellido"]}";
?>

<html>

```

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Inicio de Sesión</title>
</head>
<body bgcolor="#D2EBF7" style="font-family: Tahoma;">
  <h2>Usuario: <?php echo($nombre) ?></h2>
  <form name="form1" method="post" action="agregararticulo.php">
    <table width="282">
      <tr>
        <td width="74">Artículo</td>
        <td width="182">
          <input type="text" name="articulo" size="30" maxlength="30">
        </td>
      </tr>
      <tr>
        <td>Precio</td>
        <td>
          <input name="precio" type="text" id="precio"
            size="10" maxlength="5">
        </td>
      </tr>
      <tr>
        <td>Cantidad</td>
        <td>
          <input type="text" name="cantidad" size="10" maxlength="3">
        </td>
      </tr>
      <tr>
        <td colspan="2" align="center">
          <input type="submit" name="enviar" value="Enviar" id="enviar">
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

Inicialmente este programa presenta el siguiente formulario:



**Usuario: Gustavo Coronel**

Artículo

Precio

Cantidad

Enviar

Luego de agregar el nombre del artículo, el precio y la cantidad, agrega estos datos al carrito y luego se redirecciona el control al programa **carrito.php**.

Archivo: **cerrarsesion.php**

---

```
<?php
    session_start();
    session_unset();
    session_destroy();
    header( "location: inicio.html" );
?>
```

Después de crear todos los programas puede empezar a interactuar con ellos, debe iniciar la ejecución con el programa **inicio.html**.