

# Unidad N° 4

---

## Tecnología de desarrollo Web del lado del servidor

# Elementos básicos de PHP

---

## *Introducción*

- Con HTML y CSS es posible crear páginas web estáticas.
- Con JavaScript se pueden diseñar páginas web dinámicas de cliente.
- Con PHP se pueden diseñar páginas web dinámicas de servidor:
  - El script incluido en la página se ejecutará en el servidor web y la salida producida será enviada al cliente para que se visualice en el browser.
  - La salida puede ser un código HTML o una combinación de HTML y JavaScript

# Elementos básicos de PHP

---

## *Características del lenguaje*

- Es un lenguaje utilizado para programar scripts que se ejecutan en el servidor web, permitiendo así diseñar páginas web dinámicas de servidor.
- Brinda soporte para muchos servidores web, lo que permite que se ejecute bajo Apache, IIS, etc.
- A partir de la versión 5, tiene un gran soporte para el desarrollo de aplicaciones orientadas a objetos.
- Incluye extensiones para conectarse a varios motores de bases de datos.

# Elementos básicos de PHP

---

- Incorpora soporte para varios protocolos de Internet, como ser POP3, FTP, TELNET, etc.
  - Esto combinado con el soporte de XML forma la base para el desarrollo de web services.
- Brinda soporte para implementar canales seguros para el intercambio de datos utilizando SSL.
- Es libre, por lo cual hay libertad para utilizarlo, para modificar su código fuente y para modificar las aplicaciones desarrolladas con él.
- Es de fácil aprendizaje y su sintaxis se basa en C.
- Las aplicaciones LAMP, WAMP y XAMPP facilitan mucho la instalación y desarrollo de los scripts.

# Elementos básicos de PHP

## El primer script

```
1 <!-- Codigo_6.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Primer script PHP</title>
6 </head>
7 <body>
8 <header>
9 <?php
10 /*Este es un comentario*/
11 echo "<h1>Comenzamos la Unidad N° 4!!!</h1>";
12 ?>
13 </header>
14 <footer>
15 <p>Archivo: Codigo_6.php</p>
16 </footer>
17 </body>
18 </html>
```

Los archivos se guardan con extensión ".php"

La inclusión del código PHP se demarca con "<?php" para el inicio y ">" para el fin

Los comentarios se indican con   
/\* y \*/

Las sentencias finalizan con ";"

Es posible incluir etiquetas HTML dentro de las sentencias de impresión



# Elementos básicos de PHP

---

## *Prueba de los scripts en un software XAMPP*

1. Instalar el software XAMPP
2. Iniciar los servicios (servidor web apache, PHP, MySQL)
3. En el directorio donde se instaló el WAMP (normalmente C:\XAMPP), ubicar el subdirectorio “htdocs”
4. Guardar los scripts en ese subdirectorio o crear un nuevo subdirectorio y guardarlos allí.
5. Iniciar el navegador. En la barra de direcciones, ingresar <http://localhost/> seguido del nombre del script o del nombre del subdirectorio creado más el script a probar.

# Elementos básicos de PHP

---

## *Prueba de los scripts en un software WAMP*

1. Instalar el software WAMP
2. Iniciar los servicios (servidor web apache, PHP, MySQL)
3. En el directorio donde se instaló el WAMP (normalmente C:\WAMP), ubicar el subdirectorio “www”
4. Guardar los scripts en ese subdirectorio o crear un nuevo subdirectorio y guardarlos allí.
5. Iniciar el navegador. En la barra de direcciones, ingresar <http://localhost/> seguido del nombre del script o del nombre del subdirectorio creado más el script a probar.

# Arreglos en PHP

---

## *Introducción*

- PHP incluye la estructura de datos arreglo y la implementa con dos variantes:
  - Como arreglo enumerado: el índice es numérico y, por defecto, comienza en 0.
  - Como arreglo asociativo: el índice es una cadena de caracteres.
- En éstos arreglos los elementos no necesariamente deben ser del mismo tipo de datos.
- Además, en base a éstos, se pueden formar arreglos multidimensionales.



# Arreglos en PHP

## *Ejemplos*

- Arreglo enumerado:

\$totales	Valor	14	20	15	12
	Indice	0	1	2	3

- Arreglo asociativo:

\$claustro	Valor	125	44	22	12
	Indice	alumnos	docentes	graduados	no_docente

- Arreglo enumerado con elementos de diferente tipo:

\$de_todo	Valor	14	“vector”	5.14	[2,8,5,9]
	Indice	0	1	2	3

- \$de\_todo[3][1] = 8 (arreglo multidimensional)

# Arreglos en PHP

---

## *Arreglos superglobales de PHP*

- PHP incluye varios arreglos asociativos predefinidos que están siempre disponibles en un script:
  - `$GLOBALS`: Contiene referencias a todas las variables que definidas en el script.
  - `$_SERVER`: Es creado por el servidor web. Almacena información de las cabeceras HTTP, rutas y ubicaciones del script:
    - `PHP_SELF`
    - `SERVER_ADDR`
    - `SERVER_NAME`
    - `SERVER_PROTOCOL`
    - `REQUEST_METHOD`
    - `REQUEST_TIME`
    - `QUERY_STRING`
    - `REMOTE_ADDR`
    - `REMOTE_PORT`
    - `HTTPS`

# Arreglos en PHP

---

- `$_GET`: Contiene todas las variables enviadas desde un formulario HTML al script usando HTTP GET.
- `$_POST`: Contiene todas las variables enviadas desde un formulario HTML al script utilizando el método HTTP POST.
- `$_FILES`: Contiene información de los archivos enviados al servidor web a través del método HTTP POST (operación de upload de archivos).
- `$_COOKIE`: Contiene todas las variables enviadas al script actual usando cookies HTTP.
- `$_SESSION`: Contiene todas las variables de sesión definidas en el script. Esto permite acceder a variables definidas en otros scripts que se ejecutan en la sesión actual.

# Arreglos en PHP

## Ejemplo

```
1 <!-- Codigo_11.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Constantes y variables de PHP</title>
6 </head>
7 <body>
8 <header>
9 <h1>Constantes y variables de PHP</h1>
10 </header>
11 <section>
12 <article>
13 <?php
14 echo "Contenido de __FILE__ : ".__FILE__;
15 echo "<br>";
16 echo "Contenido de \$_SERVER['PHP_SELF'] : ".\$_SERVER['PHP_SELF'];
17 echo "<br>";
18 echo "Contenido de \$_SERVER['SERVER_ADDR'] : ".\$_SERVER['SERVER_ADDR'];
19 echo "<br>";
20 echo "Contenido de \$_SERVER['SERVER_NAME'] : ".\$_SERVER['SERVER_NAME'];
21 ?>
22 </article>
23 </section>
24 <footer>
25 <p>Archivo: Codigo_11.php</p>
26 </footer>
27 </body>
28 </html>
```



# Programación de páginas web

---

## *Introducción*

- Los formularios HTML constituyen la forma más simple para que un usuario pueda ingresar datos hacia la aplicación web.
- Luego, esos datos pueden procesarse en el cliente o en el servidor.
- Si se envían al servidor, una de las formas de procesarlos es a través de un script PHP.
- Y una de las virtudes más destacables de PHP es la facilidad con que permite recolectar la información que se envía desde un formulario HTML hacia un script y procesarla.

# Programación de páginas web

---

## *Interacción HTML – PHP*

- En un formulario HTML, el atributo “action” indica el script PHP que procesará los datos ingresados.
- Cuando los recibe, PHP guarda los datos como elementos de un arreglo asociativo que depende del método de envío del formulario. La información enviada tiene la forma nombre-valor. Así, por ejemplo, si en el formulario existe un campo de texto llamado “ciudad” y el método de envío fue POST, entonces, en el script PHP, se accederá a este valor a través de `$_POST['ciudad']`.
- Así, a cada control del formulario, le corresponde un elemento en el arreglo asociativo `$_GET` o `$_POST`.

# Programación de páginas web

## Envío de formularios HTML

Nombre

Categoría ☒ Docente ☐ Alumno

Modulos ☒ Archivo ☒ Edición

Operacion

Tablas 

Personas
Alumnos
Docentes

```
<form action="procesa.php" method="post" name="formulario">
```

```
<input name="usuario" type="text">
```

```
<input name="tipo_cat" type="radio" value="doc">
```

```
<input name="tipo_cat" type="radio" value="alu">
```

```
<input name="mod_archivo" type="checkbox" value="1">
```

```
<input name="mod_edicion" type="checkbox" value="1">
```

```
<select name="tipo_op" id="tipo_op">
```

```
<option value="0">Alta</option>
```

```
<option value="1">Baja</option>
```

```
<option value="2">Actualización</option>
```

```
</select>
```

```
<select name="tablas[]" size="3" multiple id="tablas[]">
```

```
<option value="tPersonas">Personas</option>
```

```
<option value="tAlumnos">Alumnos</option>
```

```
<option value="tDocentes">Docentes</option>
```

```
</select>
```

```
<input name="boton" type="submit" value="Enviar">
```

```
Array
```

```
(
```

```
    [usuario] => Juan
```

```
    [tipo_cat] => doc
```

```
    [mod_archivo] => 1
```

```
    [mod_edicion] => 1
```

```
    [tipo_op] => 2
```

```
    [tablas] => Array
```

```
        (
```

```
            [0] => tPersonas
```

```
            [1] => tDocentes
```

```
        )
```

```
    [boton] => Enviar
```

```
)
```

# Programación de páginas web

## *Ejemplo: Visualización de datos de formularios*

```
1 <!-- Codigo_3.htm -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Visualización de datos</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <h2>Introduzca los siguientes datos:</h2>
11 <form id="idformu" name="formu" method="post" action="Codigo_3.php">
12 Nombre: <input id="idtxtNombre" name="txtNombre" type="text" size="30" maxlength="30">
13 Edad: <input id="idtxtEdad" name="txtEdad" type="number" size="5" maxlength="3">
14 <input type="submit" id="idbtnEnviar" name="btnEnviar" value="Enviar">
15 </form>
16 </body>
17 </html>
```

```
1 <!-- Codigo_3.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Visualización de datos</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <h2>Datos recibidos:</h2>
11 <?php
12 echo "Nombre: ".$_POST['txtNombre']."<br>";
13 echo "Edad: ".$_POST['txtEdad']."<br>";
14 ?>
15 </article>
16 </section>
17 </body>
18 </html>
```



Ejemplo - Visualización de datos

localhost/lpl/Codigo\_3.htm

Introduzca los siguientes datos:

Nombre: Juana Edad: 25 Enviar



Ejemplo - Visualización de datos

localhost/lpl/Codigo\_3.php

Datos recibidos:

Nombre: Juana  
Edad: 25

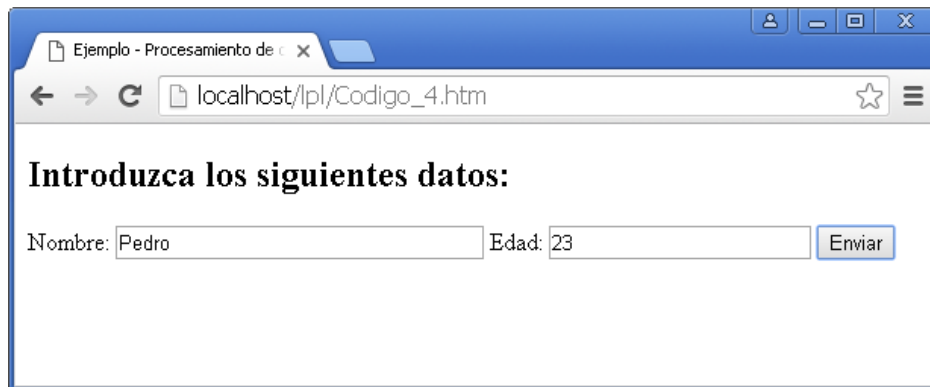


# Programación de páginas web

## Ejemplo: Procesamiento de formularios (1)

```
1 <!-- Codigo_4.htm -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Procesamiento de datos</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <h2>Introduzca los siguientes datos:</h2>
11 <form id="idformu" name="formu" method="post" action="Codigo_4.php">
12 Nombre: <input id="idtxtNombre" name="txtNombre" type="text" size="30" maxlength="30">
13 Edad: <input id="idtxtEdad" name="txtEdad" type="number" size="5" maxlength="3">
14 <input id="idbtnEnviar" type="submit" name="btnEnviar" value="Enviar">
15 </form>
16 </article>
17 </section>
18 </body>
19 </html>
```

```
1 <!-- Codigo_4.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Procesamiento de datos</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <h2>Datos recibidos:</h2>
11 <?php
12 define ("canti_dias",365);
13 echo "Nombre: ".$_POST['txtNombre']."<br>";
14 echo "Edad: ".$_POST['txtEdad']."<br>";
15 echo "Has vivido ".$_POST['txtEdad']*canti_dias." días!!";
16 ?>
17 </article>
18 </section>
19 </body>
20 </html>
```



Ejemplo - Procesamiento de datos

localhost/lpl/Codigo\_4.htm

Introduzca los siguientes datos:

Nombre: Pedro Edad: 23



Ejemplo - Procesamiento de datos

localhost/lpl/Codigo\_4.php

Datos recibidos:

Nombre: Pedro  
Edad: 23  
Has vivido 8395 días!!

# Programación de páginas web

## *Ejemplo: Procesamiento de formularios (2)*

```
1 <!-- Codigo_112.htm -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Procesamiento de datos</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <h2>Introduzca los siguientes datos:</h2>
11 <form id="idformu" name="formu" method="post" action="Codigo_112.php">
12 <table width="100%" border="0" cellpadding="0">
13   <tr>
14     <td width="10%">Tratamiento:</td>
15     <td width="90%">
16       <select id="idcmbTratamiento" name="cmbTratamiento">
17         <option value="0" selected>---</option>
18         <option value="Sr.">Sr.</option>
19         <option value="Sra.">Sra.</option>
20       </select>
21     </td>
22   </tr>
23   <tr>
24     <td>Nombre:</td>
25     <td><input id="idtxtNombre" name="txtNombre" type="text"></td>
26   </tr>
27   <tr>
28     <td>Edad:</td>
29     <td><input id="idtxtEdad" name="txtEdad" type="number"></td>
30   </tr>
31   <tr>
32     <td colspan="2"><input id="idbtnEnviar" type="submit" name="btnEnviar" value="Enviar"></td>
33   </tr>
34 </table>
35 </form>
36 </article>
37 </section>
38 </body>
39 </html>
```

# Programación de páginas web

## *Ejemplo (continuación)*

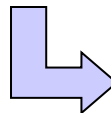
```
1 <!-- Codigo_112.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Procesamiento de datos</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <h2>Datos recibidos:</h2>
11 <?php
12 define ("canti_dias",365);
13 if (($_POST['cmbTratamiento']!='0') && ($_POST['txtNombre']!='') && ($_POST['txtEdad']!=''))
14 {
15     echo "Bienvenido ".$_POST['cmbTratamiento']." ".$_POST['txtNombre']. "!!<br>";
16     echo "Su edad es ".$_POST['txtEdad']. "<br>";
17     echo "Ud ha vivido ".$_POST['txtEdad']*canti_dias." días!!";
18 }
19 else
20 {
21     echo 'Faltan datos para procesar.';
22 }
23 echo '<br><a href="Codigo_112.htm">Volver al inicio</a>';
24 ?>
25 </article>
26 </section>
27 </body>
28 </html>
```

# Programación de páginas web

## *Ejemplo (continuación)*



A screenshot of a web browser window titled 'Ejemplo - Procesamiento de'. The address bar shows 'localhost/lpl/Codigo\_112.htm'. The page content includes the heading 'Introduzca los siguientes datos:' followed by a form with three input fields: 'Tratamiento:' with a dropdown menu showing 'Sr', 'Nombre:' with the text 'Nestor', and 'Edad:' with the text '40'. Below these fields is an 'Enviar' button.



A screenshot of a web browser window titled 'Ejemplo - Procesamiento de'. The address bar shows 'localhost/lpl/Codigo\_112.php'. The page content includes the heading 'Datos recibidos:' followed by the text 'Bienvenido Sr. Nestor!!', 'Su edad es 40', and 'Ud ha vivido 14600 días!!'. At the bottom is a link labeled 'Volver al inicio'.

# Programación de páginas web

---

## *Página autoprocesada*

- Hasta el momento, hemos trabajado el procesamiento de datos mediante el desarrollo de:
  - Una página HTML para ingresar los datos
  - Un script PHP para procesarlos
- Esta técnica puede mejorarse utilizando un único script PHP que se encargue de realizar las dos acciones.
- Para esto, el script PHP puede tener una estructura básica y simple:
  - si hay datos para procesar, se procesan, y sino, se solicitan.

# Programación de páginas web

## pagina.htm

```
<!DOCTYPE html>
<html>
.....
.....
<form id="form" name="form" method="post"
action="procesa.php">
Nombre: <input name="nombre" type="text">
Edad: <input name="edad" type="text">
<input type="submit" name="boton" value="Enviar">
</form>
.....
.....
</html>
```



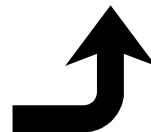
## procesa.php

```
<!DOCTYPE html>
<html>
.....
<?php
define ("canti_dias",365);
echo "Nombre: ".$_POST['nombre'];
echo "Edad: ".$_POST['edad'];
echo "Dias vividos ".$_POST['edad']*canti_dias;

?>
.....
</html>
```

## procesa.php

```
<!DOCTYPE html>
<html>
.....
<?php
if ($_POST['nombre']!="" && $_POST['edad']!="")
{
    define ("canti_dias",365);
    echo "Nombre: ".$_POST['nombre'];
    echo "Edad: ".$_POST['edad'];
    echo "Dias vividos ".$_POST['edad']*canti_dias;
    echo '<a href="procesa.php">Volver</a>';
}
else
{
    ?>
<form id="form" name="form" method="post"
action="procesa.php">
Nombre: <input name="nombre" type="text">
Edad: <input name="edad" type="text">
<input type="submit" name="boton" value="Enviar">
</form>
<?php
}
?>
.....
</html>
```



# Programación de páginas web

## Ejemplo: Página autoprocesada (1)

```
1 <!-- Codigo_122.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Página autoprocesada</title>
6 </head>
7 <body>
8 <?php
9     if ($_POST['nombre']!=" " && $_POST['edad']!=" ")
10     {
11         define ("canti_dias",365);
12         echo "Nombre: ".$_POST['nombre']."<br><br>";
13         echo "Edad: ".$_POST['edad']."<br><br>";
14         echo "Has vivido ".$_POST['edad']*canti_dias." días!!<br>";
15         echo "<br><a href='Codigo_122.php'>Volver al inicio</a>";
16     }
17     else
18     {
19     ?>
20     <section>
21         <article>
22             <h2>Introduzca los siguientes datos:</h2>
23             <form id="idform" name="form" method="post" action="Codigo_122.php">
24                 Nombre: <input id="idnombre" name="nombre" type="text" size="30" maxlength="30">
25                 Edad: <input id="idedad" name="edad" type="text" size="5" maxlength="3">
26                 <input type="submit" id="idsubmit" name="Submit" value="Enviar">
27             </form>
28         </article>
29     </section>
30 <?php
31     }
32 ?>
33 </body>
34 </html>
```

Nombre: Maria

Edad: 32

Has vivido 11680 días!!

[Volver al inicio](#)

**Introduzca los siguientes datos:**

Nombre:  Edad:

# Programación de páginas web

## Ejemplo: Página autoprocesada (2)

```
1 <!-- Codigo_5.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Página autoprocesada</title>
6 </head>
7 <body>
8 <?php
9 if (isset($_POST['txtNombre']) && isset($_POST['txtEdad']))
10 { include_once("Codigo_5.class.php");
11   $usuario = new persona($_POST['txtNombre'], $_POST['txtEdad']);
12   echo "<h2>Datos recibidos:</h2>";
13   echo "Nombre: ".$usuario->getNombre()."<br>";
14   echo "Edad: ".$usuario->getEdad()."<br>";
15   echo "Has vivido ".$usuario->getDiasVividos()." días!!"; }
16 else
17 {
18 <?>
19 <h2>Introduzca los siguientes datos:</h2>
20 <form id="idfrmPpal" name="frmPpal" method="post" action="Codigo_5.php">
21   <p>Nombre: <input id="idtxtNombre" name="txtNombre" type="text" size="30"></p>
22   <p>Edad: <input id="idtxtEdad" name="txtEdad" type="number" size="5">
23   <input id="idbtnEnviar" type="submit" name="btnEnviar" value="Enviar"></p>
24 </form>
25 <?php
26 }
27 <?>
28 </body>
29 </html>
```

```
1 <?php
2 //Codigo_5.class.php
3 class persona
4 {
5     private $nombre;
6     private $edad;
7
8     public function __construct($PNombre, $PEdad)
9     { $this->nombre = $PNombre;
10       $this->edad = $PEdad; }
11
12     public function getNombre()
13     { return $this->nombre; }
14
15     public function setNombre($PNombre)
16     { $this->nombre = $PNombre; }
17
18     public function getEdad()
19     { return $this->edad; }
20
21     public function setEdad($PEdad)
22     { $this->edad = $PEdad; }
23
24     public function getDiasVividos()
25     { define("cantiDiasAnio", 365);
26       return $this->getEdad()*cantiDiasAnio; }
27 }
28 <?>
```

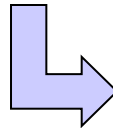


# Programación de páginas web

*Ejemplo: Página autoprocesada (2) - cont.*



A screenshot of a web browser window. The title bar says "Ejemplo - Página autoprocesada". The address bar shows "localhost/pl/Codigo\_5.php". The page content has the heading "Introduzca los siguientes datos:". Below it, there are two input fields: "Nombre:" with the value "Pedro" and "Edad:" with the value "25". To the right of the "Edad:" field is a button labeled "Enviar".



A screenshot of a web browser window. The title bar says "Ejemplo - Página autoprocesada". The address bar shows "localhost/pl/Codigo\_5.php". The page content has the heading "Datos recibidos:". Below it, the processed data is displayed: "Nombre: Pedro", "Edad: 25", and "Has vivido 9125 días!!".

# Programación de páginas web

## *Ejemplo: Página autoprocesada (3)*

```
1 <?php
2 //Codigo_26.class.php
3 class archivo
4 {
5     private $nombre;
6     private $tipo;
7     private $tamanio;
8     private $nombre_temporal;
9
10    public function __construct($PNombre)
11    {    $this->nombre = $PNombre;    }
12
13    public function getNombre()
14    {    return $this->nombre;    }
15
16    public function setNombre($PNombre)
17    {    $this->nombre = $PNombre;    }
18
19    public function getTipo()
20    {    return $this->tipo;    }
21
22    public function setTipo($PTipo)
23    {    $this->tipo = $PTipo;    }
24
25    public function getTamanio()
26    {    return $this->tamanio;    }
27
28    public function setTamanio($PTamanio)
29    {    $this->tamanio = $PTamanio;    }
30
31    public function getNombreTemporal()
32    {    return $this->nombre_temporal;    }
33
34    public function setNombreTemporal($PNombreTemp)
35    {    $this->nombre_temporal = $PNombreTemp;    }
36
37    public function moverArchivo($PDestino)
38    {    return move_uploaded_file($this->getNombreTemporal(),$PDestino);    }
39 }
40 ?>
```

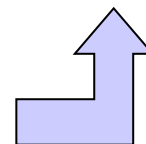
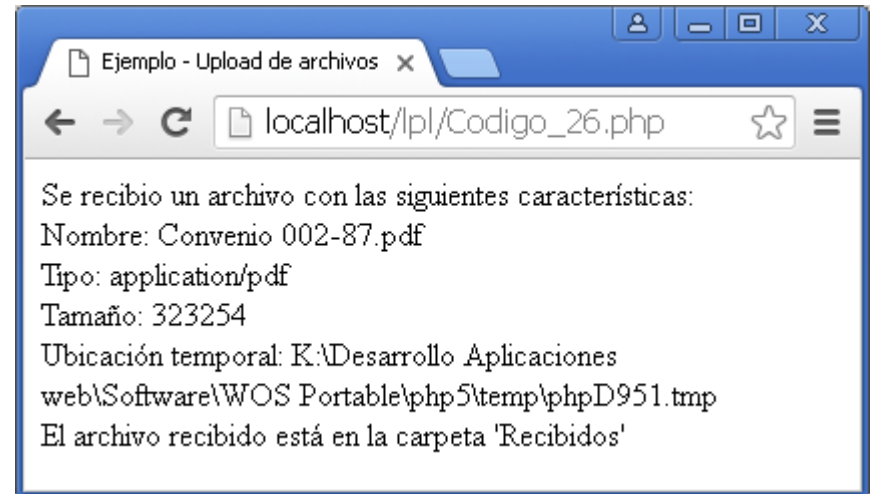
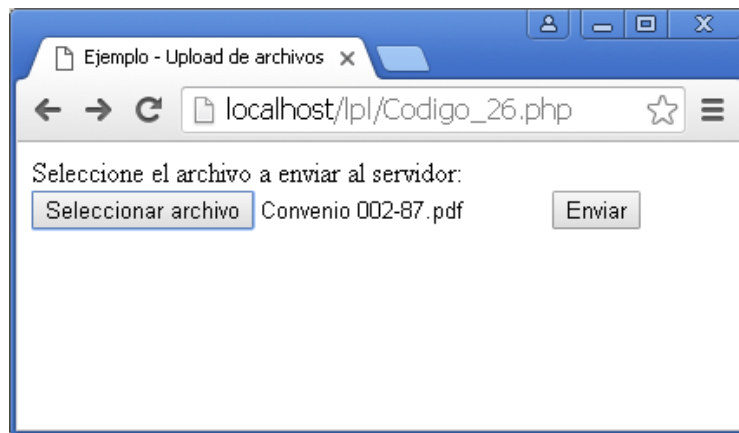
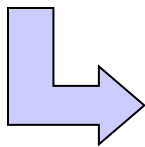
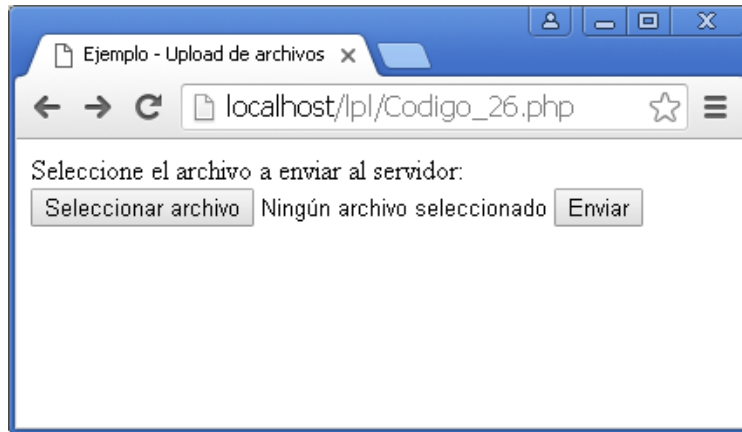
# Programación de páginas web

## *Ejemplo: Página autoprocesada (3) – cont.*

```
1 <!-- Codigo_26.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Upload de archivos</title>
6 </head>
7 <body>
8 <?php
9 if (isset($_POST['btnEnviar']) && $_POST['btnEnviar']=="Enviar")
10 { if (isset($_FILES['txtArchivo']))
11 { include_once("Codigo_26.class.php");
12 $archivoSubido = new archivo($_FILES['txtArchivo']['name']);
13 $archivoSubido->setTipo($_FILES['txtArchivo']['type']);
14 $archivoSubido->setTamaño($_FILES['txtArchivo']['size']);
15 $archivoSubido->setNombreTemporal($_FILES['txtArchivo']['tmp_name']);
16 echo "Se recibio un archivo con las siguientes características:";
17 echo "<br>Nombre: ".$archivoSubido->getNombre();
18 echo "<br>Tipo: ".$archivoSubido->getTipo();
19 echo "<br>Tamaño: ".$archivoSubido->getTamaño();
20 echo "<br>Ubicación temporal: ".$archivoSubido->getNombreTemporal();
21 $destino = "Recibidos/".$archivoSubido->getNombre();
22 if ($archivoSubido->moverArchivo($destino))
23 { echo "<br>El archivo recibido está en la carpeta 'Recibidos'"; }
24 else
25 { echo "<br>Error. No fue posible guardar el archivo enviado"; }
26 }
27 else
28 { echo "No se recibio el archivo"; }
29 }
30 else
31 {
32 ?>
33 <form action="<?php echo $PHP_SELF; ?>" method="post" enctype="multipart/form-data" name="frmPpal" id="idfrmPpal">
34 Seleccione el archivo a enviar al servidor: <input id="idtxtArchivo" name="txtArchivo" type="file">
35 <input name="btnEnviar" type="submit" id="idbtnEnviar" value="Enviar">
36 </form>
37 <?php } ?>
38 </body>
39 </html>
```

# Programación de páginas web

## *Ejemplo: Página autoprocesada (3) – cont.*



# Programación de páginas web

## *Ejemplo: Página autoprocesada (4)*

```
1 <!-- Codigo_25.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo - Página autoprocesada</title>
6 <link rel="stylesheet" href="Codigo_25.css" />
7 </head>
8 <body>
9 <section>
10 <article>
11 <?php
12 if (!isset($_POST['btnConsultar']) || ($_POST['btnConsultar']!="Consultar"))
13 {
14 }
15 <h2>Consulta de vuelos</h2>
16 <form id="idfrmPpal" name="frmPpal" method="post" action="Codigo_25.php">
17 <table class="tablaFormulario">
18 <tr>
19 <td class="izquierda">Aeropuerto</td>
20 <td class="derecha">
21 <select class="cuadroTexto" name="cmbAeropuerto" id="idcmbAeropuerto">
22 <option value="1">Aeroparque</option>
23 <option value="2" selected>Comodoro Rivadavia</option>
24 <option value="3">C rdoba</option>
25 <option value="4">Mendoza</option>
26 </select>
27 </td>
28 </tr>
29 <tr>
30 <td class="izquierda">Fecha</td>
31 <td class="derecha">
32 <input class="cuadroTexto" id="idtxtFecha" name="txtFecha" type="date">
33 </td>
34 </tr>
35 <tr>
36 <td class="izquierda">Tipo de vuelo</td>
37 <td class="derecha">
38 <input id="idrdrnTipoOperacion" name="rdrnTipoOperacion" type="radio" value="0" checked>Arribos
39 <br>
40 <input id="idrdrnTipoOperacion" name="rdrnTipoOperacion" type="radio" value="1">Partidas
41 </td>
42 </tr>
```

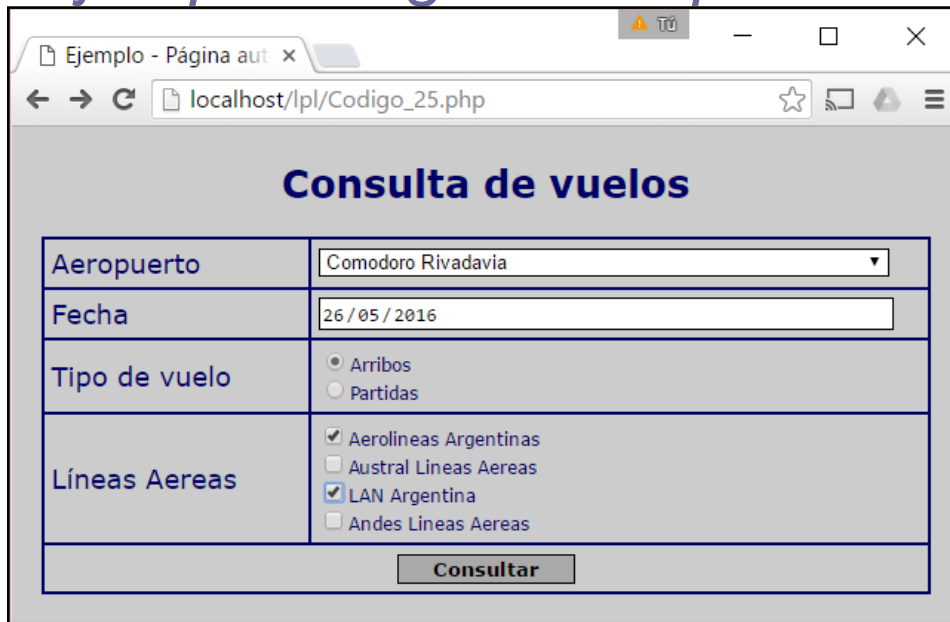
# Programación de páginas web

## Ejemplo: Página autoprocesada (4) – Continuación

```
43 <tr>
44 <td class="izquierda">Lineas Aereas</td>
45 <td class="derecha">
46 <input id="idchkAR" name="chkAR" type="checkbox" value="1" checked="">Aerolineas Argentinas<br>
47 <input id="idchkAU" name="chkAU" type="checkbox" value="1" >Austral Lineas Aereas<br>
48 <input id="idchk4M" name="chk4M" type="checkbox" value="1" >LAN Argentina<br>
49 <input id="idchkAN" name="chkAN" type="checkbox" value="1" >Andes Lineas Aereas
50 </td>
51 </tr>
52 <tr>
53 <td class="combinada" colspan="2">
54 <input class="boton" id="idbtnConsultar" name="btnConsultar" type="submit" value="Consultar"></td>
55 </tr>
56 </table>
57 </form>
58 <?php
59 }
60 else
61 {
62     $aux = array();
63     $consulta = 'SELECT * FROM vuelos ';
64     $consulta .= 'WHERE id_aeropuerto = '.$_POST['cmbAeropuerto'];
65     $consulta .= ' AND fecha = "'.$_POST['txtFecha'].'"';
66     $consulta .= ' AND tipo_operacion = '.$_POST['rdnTipoOperacion'];
67     if ($_POST['chkAR'])
68     {
69         $aux[] = 'linea = 1';
70     }
71     if ($_POST['chkAU'])
72     {
73         $aux[] = 'linea = 2';
74     }
75     if ($_POST['chk4M'])
76     {
77         $aux[] = 'linea = 3';
78     }
79     if ($_POST['chkAN'])
80     {
81         $aux[] = 'linea = 4';
82     }
83     $consulta .= " AND (" . implode(" OR ", $aux) . ")";
84     echo "<h2>Consulta a realizar:</h2>";
85     echo "<p>".$_POST['txtFecha']."</p>";
86 }
87 >
88 </article>
89 </section>
90 </body>
91 </html>
```

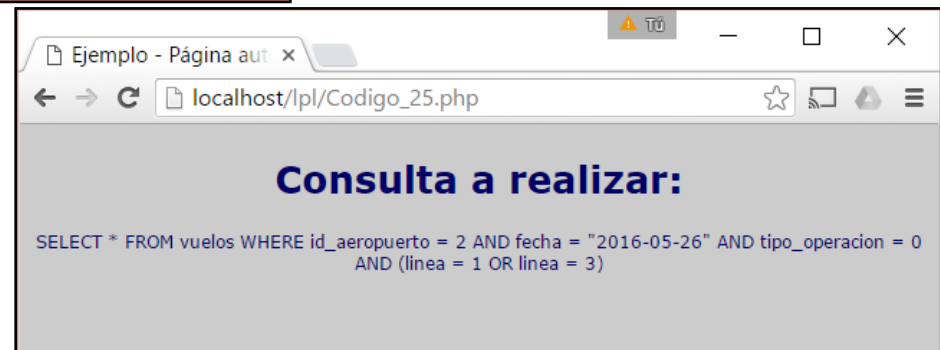
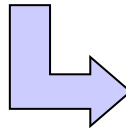
# Programación de páginas web

## *Ejemplo: Página autoprocesada (4) - Continuación*



The screenshot shows a web browser window with the address bar displaying 'localhost/lpl/Codigo\_25.php'. The page title is 'Ejemplo - Página aut x'. The main content area is titled 'Consulta de vuelos' and contains a form with the following fields:

Aeropuerto	Comodoro Rivadavia
Fecha	26/05/2016
Tipo de vuelo	<input type="radio"/> Arribos <input type="radio"/> Partidas
Líneas Aereas	<input checked="" type="checkbox"/> Aerolineas Argentinas <input type="checkbox"/> Austral Líneas Aereas <input checked="" type="checkbox"/> LAN Argentina <input type="checkbox"/> Andes Líneas Aereas
<input type="button" value="Consultar"/>	



The screenshot shows the same web browser window, but the main content area now displays the SQL query generated by the form:

**Consulta a realizar:**

```
SELECT * FROM vuelos WHERE id_aeropuerto = 2 AND fecha = "2016-05-26" AND tipo_operacion = 0  
AND (linea = 1 OR linea = 3)
```

# Programación de páginas web

---

## *Cuestiones de seguridad para el ingreso de datos*

- Es importante validar todos los datos ingresados por el usuario. Esto asegura que los valores recibidos por el script son los esperados.
- Como regla, cualquier dato ingresado por el usuario debe suponerse como erróneo, por lo cual debe validarse antes de utilizarse.
- Si el usuario ingresó código HTML en un cuadro de texto, el código será interpretado y sus efectos podrían ser graves.
- Para evitar esto, se pueden usar las funciones *htmlspecialchars* o *htmlentities*, que impiden que se interpreten los caracteres especiales de HTML.



# Sesiones

---

## *Ejercicio*

- Desarrollar una aplicación web que permita a un usuario ingresar una serie de valores (uno a uno, en momentos distintos) y, a medida que se ingresa, se debe mostrar cuántos elementos ya fueron ingresados y la lista de ellos.

# Sesiones

---

## *Introducción*

- Supóngase que debemos desarrollar una aplicación web que le permita a un usuario ingresar una serie de valores (uno a uno, en momentos distintos) y, a medida que se ingresa, se debe mostrar cuántos elementos ya fueron ingresados y la lista de ellos.
- Una primera implementación sería la de definir un arreglo en el script e ir guardando en él los valores ingresados:

# Sesiones

## *Primera implementación*

```
1  <?php
2  //
3  class vector_enumerado
4  {   private $elementos;
5
6      public function __construct()
7      {   $this->elementos = array();
8          for($i=0; $i<func_num_args(); $i++)
9              {   $this->elementos[] = func_get_arg($i);   }
10     }
11
12     public function getElementos()
13     {   return $this->elementos;   }
14
15     public function setElementos($PArreglo)
16     {   $this->elementos = $PArreglo;   }
17
18     public function totalElementos()
19     {   return count($this->elementos);   }
20
21     public function agregarElemento($PElemento)
22     {   $this->elementos[] = $PElemento;   }
23
24     public function imprimirElementos()
25     {   return implode(", ", $this->getElementos());   }
26 }
27 ?>
```

# Sesiones

## Primera implementación (continuación)

```
1 <!-- Codigo_42.php -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5 <title>Ejemplo de introducción a sesiones - 1</title>
6 </head>
7 <body>
8 <section>
9 <article>
10 <h2>Introducción a sesiones - Ejemplo 1:</h2>
11 <?php
12 function cargaClase($clase) {
13     require_once("Codigo_42.".$clase.".class.php");
14 }
15 spl_autoload_register('cargaClase');
16 if (isset($_POST['txtValor']))
17 {     echo "<br>Valor recibido: ". $_POST['txtValor']."<br>";
18     if (!isset($vector))
19     {     $vector = new vector_enumerado();     }
20     $vector->agregarElemento($_POST['txtValor']);
21     echo "Lista recibida: ".$vector->imprimirElementos()."<br>";
22     echo "Total de elementos recibidos: ".$vector->totalElementos()."<br>";
23     ?>
24     <input id="idbuttonVolver" name="btnVolver" type="submit" value="Otro valor" onClick="parent.location='Codigo_42.php';">
25 <?php
26 } else
27 {
28     ?>
29     <form action="Codigo_42.php" method="post" name="formulario" id="idformulario">
30     Ingrese un valor:<input name="txtValor" type="number" id="idtxtValor">
31     <input name="btnEnviar" type="submit" id="idbuttonEnviar" value="Enviar">
32     </form>
33 <?php     }     ?>
34 </article>
35 </section>
36 </body>
37 </html>
```

# Sesiones

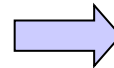
## *Primera implementación (continuación)*

Ejemplo de introducción a se: x

localhost/lpl/Codigo\_42.php

**Introducción a sesiones - Ejemplo 1:**

Ingrese un valor:



Ejemplo de introducción a se: x

localhost/lpl/Codigo\_42.php

**Introducción a sesiones - Ejemplo 1:**

Valor recibido: 25  
Lista recibida: 25  
Total de elementos recibidos: 1



Ejemplo de introducción a se: x

localhost/lpl/Codigo\_42.php

**Introducción a sesiones - Ejemplo 1:**

Ingrese un valor:



Ejemplo de introducción a se: x

localhost/lpl/Codigo\_42.php

**Introducción a sesiones - Ejemplo 1:**

Valor recibido: 33  
Lista recibida: 33  
Total de elementos recibidos: 1

# Sesiones

---

- Se observa que los valores ingresados se pierden en cada entrada que hace el usuario.
- Esto es porque las variables utilizadas en los scripts PHP **nacen y mueren** con la ejecución del script.
- Entonces, surgen problemas cuando es necesario mantener el valor de una variable durante varias ejecuciones del mismo script o entre algunos o todos los que conforman la aplicación web.
- Para solucionar esto, se puede utilizar el arreglo `$_SESSION`, que permite guardar todas aquellas variables que deben mantener su valor en una parte o en todos los scripts que componen la aplicación.

# Sesiones

---

- Así, el arreglo superglobal `$_SESSION` contiene todas las variables definidas por el programador que deben ser accesibles en uno, una parte o todos los scripts del sitio.
- Para programar con el arreglo `$_SESSION`, PHP incluye varias funciones y algunas son:

Función	Descripción
<code>Session_start</code>	Inicia una sesión o la retoma si ya estuviera iniciada.
<code>Session_destroy</code>	Destruye una sesión.
<code>Session_id</code>	Devuelve el id de la sesión actual (código alfanumérico que identifica unívocamente a la sesión actual).

# Sesiones

---

## *Forma de trabajo con sesiones*

- Para trabajar con el arreglo `$_SESSION` dentro de un script, el primer paso es invocar a la función *session\_start*.
- Luego, en `$_SESSION` se guardan todas las variables de sesión necesarias. Los posibles elementos de este arreglo son:
  - Valores lógicos
  - Números (enteros o reales)
  - Cadenas
  - Arreglos
  - Objetos



# Sesiones

---

- A la acción de insertar un elemento en `$_SESSION` se lo conoce como registrar una variable de sesión.
- Una vez que la variable fue registrada, estará disponible en cualquier script que trabaje con la sesión.
- Finalmente, para terminar la sesión, es necesario destruirla. Para esto, se debe:
  - “limpiar” el arreglo `$_SESSION` (asignándole un nuevo arreglo vacío), y
  - eliminar la sesión (invocando a la función *`session_destroy`*).

# Sesiones

## Segunda implementación (solución de la primera)

```
1 <!-- Codigo_43.php -->
2 <?php
3 function cargaClase($clase) {
4     include("Codigo_42.".$clase.".class.php");
5 }
6 spl_autoload_register('cargaClase');
7 session_start();
8 ?>
9 <!DOCTYPE html>
10 <html lang="es">
11 <head>
12 <title>Ejemplo de introducción a sesiones - 2</title>
13 </head>
14 <body>
15 <section>
16 <article>
17 <h2>Introducción a sesiones - Ejemplo 2:</h2>
18 <?php
19 if (isset($_POST['txtValor']))
20 {
21     echo "<br>Valor recibido: " . $_POST['txtValor'] . "<br>";
22     if (!isset($_SESSION['vector']))
23     {
24         $vector = new vector_enumerado();
25     }
26     else
27     {
28         $vector = $_SESSION['vector'];
29     }
30     $vector->agregarElemento($_POST['txtValor']);
31     echo "Lista recibida: " . $vector->imprimirElementos() . "<br>";
32     echo "Total de elementos recibidos: " . $vector->totalElementos() . "<br>";
33     $_SESSION['vector'] = $vector;
34     ?>
35     <input id="idbuttonVolver" name="btnVolver" type="button" value="Otro valor" onClick="parent.location='Codigo_43.php';">
36     <input id="idbuttonCerrar" name="btnCerrar" type="button" value="Finalizar" onClick="parent.location='Codigo_44.php';">
37 <?php
38 } else {
39 ?>
40 <form action="Codigo_43.php" method="post" name="formulario" id="idformulario">
41 <div>
42 <div>
43 Ingrese un valor:<input name="txtValor" type="number" id="idtxtValor">
44 <input name="btnEnviar" type="submit" id="idbuttonEnviar" value="Enviar">
45 </div>
46 </div>
47 </form>
48 <?php } ?>
49 </article>
50 </section>
51 </body>
52 </html>
```

# Sesiones

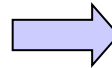
## *Segunda implementación (continuación)*

Ejemplo de introducción a se: x

localhost/lpl/Codigo\_43.php

**Introducción a sesiones - Ejemplo 2:**

Ingrese un valor:



Ejemplo de introducción a se: x

localhost/lpl/Codigo\_43.php

**Introducción a sesiones - Ejemplo 2:**

Valor recibido: 41  
Lista recibida: 41  
Total de elementos recibidos: 1

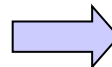


Ejemplo de introducción a se: x

localhost/lpl/Codigo\_43.php

**Introducción a sesiones - Ejemplo 2:**

Ingrese un valor:



Ejemplo de introducción a se: x

localhost/lpl/Codigo\_43.php

**Introducción a sesiones - Ejemplo 2:**

Valor recibido: 12  
Lista recibida: 41, 12  
Total de elementos recibidos: 2

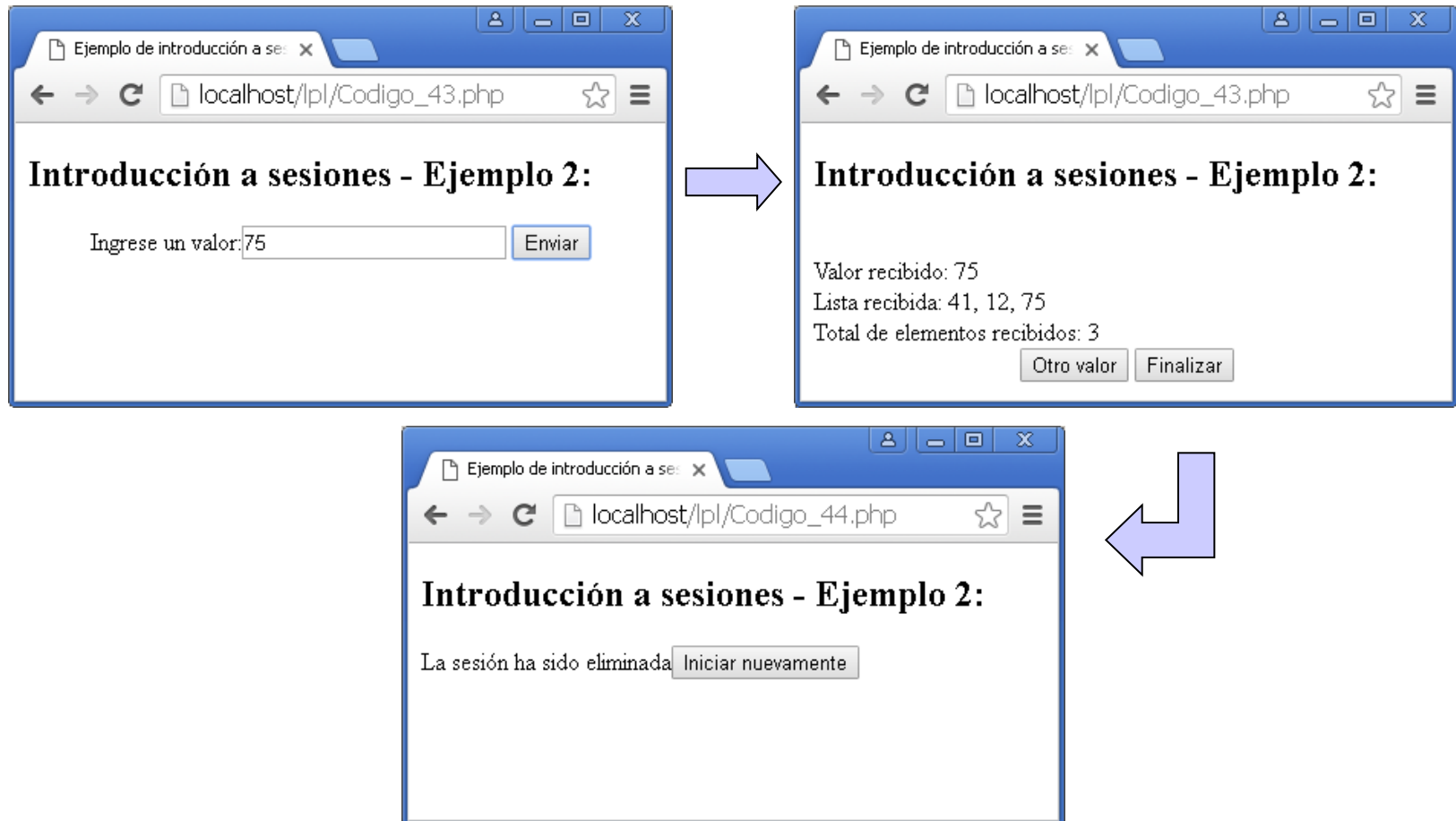
# Sesiones

## Segunda implementación (continuación – Destrucción de la sesión)

```
1 <!-- Codigo_44.php -->
2 <?php session_start(); ?>
3 <!DOCTYPE html>
4 <html lang="es">
5 <head>
6 <title>Ejemplo de introducción a sesiones - 2</title>
7 </head>
8 <body>
9 <section>
10 <article>
11 <?php
12 echo "<h2>Introducción a sesiones - Ejemplo 2:</h2>";
13 if (isset($_SESSION['vector']))
14 {   $_SESSION = array();
15     session_destroy();
16     echo "La sesión ha sido eliminada";
17 }
18 else
19 {   echo "No hay una sesión iniciada";
20 }
21 ?>
22 <input id="idbtnVolver" name="btnVolver" type="button" value="Iniciar nuevamente" onClick="parent.location='Codigo_43.php';">
23 </article>
24 </section>
25 </body>
26 </html>
```

# Sesiones

## *Segunda implementación (continuación)*



# Sesiones

---

## *Aplicación del arreglo \$\_SESSION*

- El arreglo \$\_SESSION puede utilizarse para desarrollar aplicaciones web que trabajen con sesiones.
- Conceptualmente, una sesión comienza cuando un usuario ingresa a la aplicación web y termina cuando la abandona.
- En algunas aplicaciones, la sesión comienza cuando el usuario se identifica (logueo) y termina cuando el usuario abandona la aplicación (salir).
- Durante ese tiempo, se pueden usar variables que indican que el usuario está dentro de la sesión, y de esa forma, puede acceder a ciertas funcionalidades.

# Cookies

---

## *Introducción*

- Una cookie es un recurso que permite al servidor web guardar información en un cliente. En si, es un archivo de texto almacenado en el cliente, al que sólo puede acceder y procesar el servidor web.
- Así, este recurso permite a los programadores guardar el estado de ciertas variables, a fin de poder utilizarlas en futuros accesos de los usuarios al sitio web.
- Un ejemplo de uso de cookies se observa en los sistemas webmail, o los contadores de visitas a los sitios web.

# Cookies

---

- Si éstos sistemas utilizan cookies, cuando el usuario ingresa al sitio web, éstas se envían al servidor web para que sean utilizadas en el script. Si no hubieran cookies guardadas en el cliente, entonces el servidor web las crea y guarda para que sean utilizadas en el próximo ingreso.
- Así, el primer paso para trabajar con las cookies es crearlas. Para esto, se utiliza la función *setcookie*.
- Esta función, además de crearla, permite modificar el valor y borrarlas.
- Luego, las cookies serán accesibles a través del arreglo asociativo `$_COOKIE`.



# Cookies

---

## *Inicialización, modificación y eliminación de Cookies*

- En PHP, las cookies se crean con la función *setcookie*, la cual tiene los siguientes parámetros:
  - Nombre: es el identificador de la cookie
  - Valor: es el valor con que se guarda la cookie.
  - Expiración: es un timestamp que representa la fecha de expiración.
  - Path: es el directorio donde se ubican los scripts que tienen acceso a la cookie. Si no se especifica, se toma el directorio donde está el script que la crea.
  - Dominio: especifica el servidor o el dominio donde la cookie es válida.
  - Seguridad: indica si la cookie se transmite con HTTPS

# Cookies

---

- Todos los parámetros de la función *setcookie* son optativos, excepto el nombre.
- Algo importante: la creación de una cookie debe realizarse antes de producir cualquier salida con el script.
- Para modificar el valor de una cookie, se debe invocar a la función *setcookie* especificando el nuevo valor en el segundo parámetro.
- Para eliminar una cookie del cliente, se debe invocar a la función *setcookie* solo con el parámetro nombre.
- Otra opción es invocarla con el nombre, el valor NULL (o una cadena vacía) y una fecha de expiración anterior a la fecha actual del servidor.

# Cookies

---

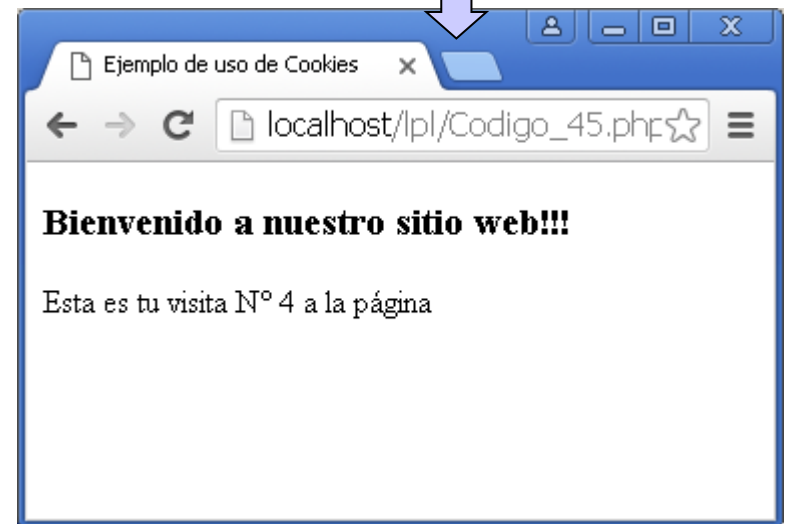
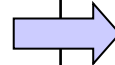
## *Acceder a los valores de las Cookies*

- Las cookies creadas por el servidor web pueden ser accedidas a través del arreglo `$_COOKIE`.
- En este arreglo, los índices son los nombres de las cookies y los elementos son los valores con que fueron guardadas en el cliente.
- Así, si en un script se creo la cookie “contador”, entonces su valor será recuperado a través del elemento `$_COOKIE['contador']`.
- Además, desde este arreglo es posible verificar la existencia de una cookie.

# Cookies

## Ejemplo

```
1 <!-- Codigo_45.php -->
2 <?php
3 $conta = 1;
4 if (isset($_COOKIE['contador']))
5 {   $conta = $_COOKIE['contador'] + 1; }
6 $expira = mktime(23,59,59,12,31,2020);
7 setcookie("contador"); //Borro la cookie antigua
8 setcookie("contador",$conta,$expira); //Creo una nueva cookie
9 ?>
10 <!DOCTYPE html>
11 <html lang="es">
12 <head>
13 <title>Ejemplo de uso de Cookies</title>
14 </head>
15 <body>
16 <section>
17 <article>
18 <?php
19 echo "<h1>Bienvenido a nuestro sitio web!!!</h1>";
20 if ($conta == 1)
21 {   echo "Esta es tu primera visita a la página"; }
22 else
23 {   echo "Esta es tu visita N° ".$conta." a la página"; }
24 ?>
25 </article>
26 </section>
27 </body>
28 </html>
```



# Cookies

---

## *Cuestiones a considerar en el uso de Cookies*

- La creación y posterior acceso a una cookie se produce en peticiones HTTP diferentes. Por esto, no es posible crear una cookie e inmediatamente acceder a su valor en una misma carga del script.
- Las cookies pueden ser un recurso interesante para los programadores, pero su uso depende siempre de la configuración del navegador del cliente.
  - Así, un sistema puede funcionar íntegramente con cookies pero si el cliente no las tiene habilitadas en su navegador, este recurso no podrá utilizarse.
- Un servidor web sólo tiene acceso a aquellas cookies que él ha creado.