

# PROGRAMACIÓN II



## SEMANA 02 VARIABLES EXTERNAS

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

## Variables Externas

Los diferentes documentos que conforman una aplicación Web necesitan comunicarse entre sí, una de las formas es utilizando formularios HTML que envían datos a un programa PHP, éstos campos enviados se convierten en variables dentro el programa PHP, y como provienen de otro documento se les denomina Variables Externas.

### Índice

<b>1</b>	<b>FORMULARIOS HTML (GET Y POST) .....</b>	<b>4</b>
<b>2</b>	<b>DIRECTIVA REGISTER_GLOBALS .....</b>	<b>6</b>
<b>3</b>	<b>ARREGLOS GLOBALES HTTP .....</b>	<b>8</b>
3.1	ARREGLO \$_POST .....	8
3.2	ARREGLO \$_GET .....	11
<b>4</b>	<b>PROGRAMAS RECURSIVOS .....</b>	<b>14</b>
<b>5</b>	<b>COOKIES HTTP.....</b>	<b>17</b>

## 1 Formularios HTML (GET y POST)

---

Los diferentes documentos que conforman una aplicación Web necesitan comunicarse entre sí, una de las formas es utilizando formularios HTML que envían datos a un programa PHP, éstos campos enviados se convierten en variables dentro el programa PHP, y como provienen de otro documento se les denomina **Variables Externas**.

Esta forma sencilla que proporciona el lenguaje PHP de manejar formularios, nos permite procesar información que el usuario ingresa a través de nuestra aplicación Web.

### Sintaxis

```
<FORM METHOD="POST/GET" ACTION="destino" ACCEPT-CHARSET="UTF-8">  
  
    Controles HTML  
  
</FORM>
```

En este libro asumo que usted ya maneja HTML y Java Script, en todo caso recomiendo conseguir un manual y leer sobre estos lenguajes.

### Ejemplo 1

Archivo: prog2\sem02\ejm01.html

---

```
<h1> Registro de Clientes</h1>  
<form method="POST" action="ejm01.php" accept-charset="UTF-8">  
    Nombre  
    <input type="text" name="txtnombre" size="20" maxlength="20"><br>  
    Email  
    <input type="text" name="txtemail" size="20" maxlength="20"><br>  
    <input type="submit" value="Enviar">  
    <input type="reset" value="Limpiar">  
</form>
```

El resultado es:



Los campos del formulario están automáticamente disponibles dentro del programa PHP, el nombre de estas variables toman el siguiente formato:

```
$NombreDelCampo
```

El uso de este método sólo es posible si la directiva **REGISTER\_GLOBALS** está en ON.

Según el método que utilicemos (POST o GET) tenemos dos arreglos globales (\$\_POST y \$\_GET) que nos permiten recibir los campos que son enviados desde el formulario, el uso de éstos arreglos es más seguro porque no depende del estado de la directiva REGISTER\_GLOBALS.

Si utilizamos el método GET debemos utilizar el siguiente formato:

```
$_GET["NombreDelCampo"]
```

Y si utilizamos el método POST es el siguiente formato:

```
$_POST["NombreDelCampo"]
```

Si el campo puede ser pasado indistintamente utilizando el método GET o POST, podemos utilizar el arreglo global \$\_REQUEST, éste arreglo no depende de la directiva REGISTER\_GLOBALS y tampoco del método de envío, el formato para obtener el valor de los campos es el siguiente:

```
$_REQUEST["NombreDelCampo"]
```

## 2 Directiva REGISTER\_GLOBALS

---

Al diseñar un formulario debemos indicar la página PHP que procesará el formulario, así como el método por el que se le pasará la información a la página, tal como se aprecia en el Ejemplo 2.

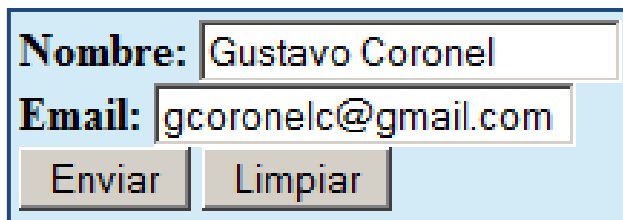
### Ejemplo 2

Archivo: prog2\sem02\ejm02.html

---

```
<body bgcolor="#D2EBF7">
  <form method=post action="ejm03.php" accept-charset="UTF-8">
    <b>Nombre: </b>
    <input type="text" name="nombre" size="20" maxlength="20" ><br>
    <b>Email: </b>
    <input type="text" name="email" size="20" maxsize="20"><br>
    <input type="submit" value="Enviar">
    <input type="reset" value="Limpiar">
  </form>
</body>
```

El resultado es el siguiente:



Cuando se envía un formulario HTML a un programa PHP, los campos de dicho formulario pasan a estar automáticamente disponibles en el programa como variables.

Es necesario revisar la directiva REGISTER\_GLOBALS en el archivo **php.ini**, esta directiva debe estar habilitada (ON) para que PHP proceda a crear las variables externas de manera automática.

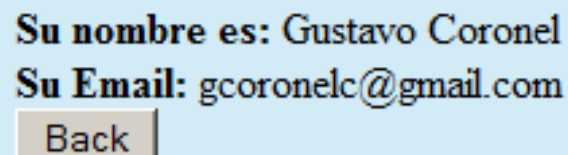
### Ejemplo 3

Archivo: prog2\sem02\ejm03.php

---

```
<?php header('Content-Type: text/html; charset=utf-8' ); ?>
<body bgcolor="#D2EBF7">
  <?php
    echo("<b>Su nombre es:</b> $nombre<br>");
    echo("<b>Su Email:</b> $email<br>");
  ?>
  <input type="button" value="Back" onClick="history.back()">
</body>
```

El resultado es el siguiente:



**Su nombre es:** Gustavo Coronel  
**Su Email:** gcoronelc@gmail.com

## 3 Arreglos Globales HTTP

---

### 3.1 Arreglo \$\_POST

Se trata de un arreglo asociativo de variables pasadas al programa actual a través del método HTTP POST.

Esta es una variable **superglobal**, esto simplemente quiere decir que está disponible en todos los contextos a lo largo del programa.

El Ejemplo 4 (ejm04.html) muestra un formulario que permite el ingreso de datos para calcular el sueldo de un trabajador, y el Ejemplo 5 (ejm05.php) recibe los datos y realiza el cálculo del sueldo.

#### Ejemplo 4

Archivo: prog2\sem02\ejm04.html

---

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body bgcolor="#D2EBF7">
  <h1>Cálculo de Sueldo</h1>
  <form method="post" action="ejm05.php" accept-charset="UTF-8">
    <table width="317">
      <tr>
        <td width="125"><b>Días Trabajados:</b></td>
        <td width="180"><input type="text" name="dt"></td>
      </tr>
      <tr>
        <td><b>Horas por Día:</b></td>
        <td><input type="text" name="hd"></td>
      </tr>
      <tr>
        <td><b>Costo por Hora:</b></td>
        <td><input type="text" name="ch"></td>
      </tr>
      <tr>
        <td colspan="2">
          <input type="submit" value="Enviar">
          <input type="reset" value="Limpiar">
        </td>
      </tr>
    </table>
```

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

El resultado es el siguiente:

## Cálculo de Sueldo

**Días Trabajados:**

**Horas por Día:**

**Costo por Hora:**

### Ejemplo 5

Archivo: Prog2\sem02\ejm05.php

```
<?php
// Datos
$dt = $_POST["dt"]; // Días trabajados
$hd = $_POST["hd"]; // Horas por día
$ch = $_POST["ch"]; // Costo po hora
// Proceso
$total = $dt * $hd * $ch;
$ir = $total * 0.10; # Impuesto a la renta
$neto = $total - $ir;
// Formatos
$ch = "S/. " . number_format($ch, 2, ".", "");
$total = "S/. " . number_format($total, 2, ".", "");
$ir = "S/. " . number_format($ir, 2, ".", "");
$neto = "S/. " . number_format($neto, 2, ".", "");
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script>
function boton_back(){
    document.location.href = "ejm04.html";
}
</script>
```



```
</head>
<body bgcolor="#D2EBF7">
  <h1>Cálculo de Sueldo</h1>
  <table border=1 width=241>
    <tr>
      <th align=center colspan=2 border=0>Datos</th>
    </tr>
    <tr>
      <td width=53%><b>Días trabajados</b></td>
      <td width="47%"><?php echo($dt) ?></td>
    </tr>
    <tr>
      <td><b>Horas por día</b></td>
      <td><?php echo($hd) ?></td>
    </tr>
    <tr>
      <td><b>Costo por hora</b></td>
      <td><?php echo($ch) ?></td>
    </tr>
    <tr>
      <th align=center colspan=2 border=0>
        Resultado
      </th>
    </tr>
    <tr>
      <td><b>Total</b></td>
      <td><?php echo($total) ?></td>
    </tr>
    <tr>
      <td><b>Impuesto Renta</b></td>
      <td><?php echo($ir) ?></td>
    </tr>
    <tr>
      <td><b>Neto</b></td>
      <td><?php echo($neto) ?></td>
    </tr>
  </table>
  <input type="button" value="Retornar" onClick="boton_back()">
</body>
</html>
```

El resultado se muestra a continuación:

Cálculo de Sueldo	
<b>Datos</b>	
Días trabajados	6
Horas por día	8
Costo por hora	S/. 35.00
<b>Resultado</b>	
Total	S/. 1,680.00
Impuesto Renta	S/. 168.00
Neto	S/. 1,512.00
<input type="button" value="Retornar"/>	

### 3.2 Arreglo \$\_GET

Se trata de un arreglo asociativo de variables pasadas al programa actual a través del método HTTP GET.

Esta es una variable **superglobal**, esto simplemente quiere decir que está disponible en todos los contextos a lo largo del programa.

El Ejemplo 6 (ejm06.php) muestra un formulario que permite el ingreso de dos números, éstos datos son enviados al programa del Ejemplo 7 (ejm07.php), éste programa recibe los datos, calcula la suma y muestra los resultados.

#### Ejemplo 6

Archivo: prog2\sem02\ejm06.html

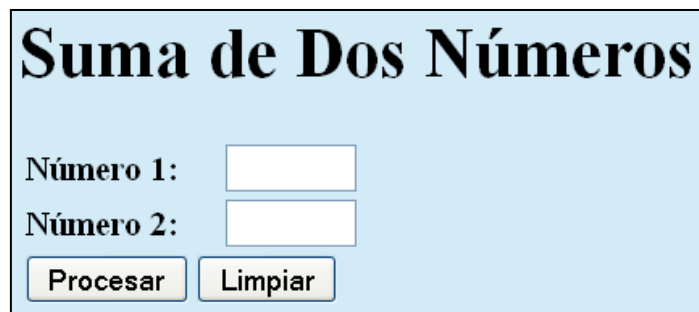
```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body bgcolor="#D2EBF7">
  <h1>Suma de Dos Números</h1>
  <form method="get" action="ejm07.php">
    <table width="165">
      <tr>
        <td width="88"><b>Número 1:</b></td>
```

```

        <td width="65">
            <input name="N1" type="text" size="6" maxlength="6">
        </td>
    </tr>
    <tr>
        <td><b>Número 2:</b></td>
        <td><input name="N2" type="text" size="6" maxlength="6"></td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" value="Procesar">
            <input type="reset" value="Limpiar"></td>
    </tr>
</table>
</form>
</body>
</html>

```

El resultado es:



## Ejemplo 7

Archivo: Prog2\sem02\ejm07.php

```

<?php
    // Datos
    $n1 = $_GET["N1"];
    $n2 = $_GET["N2"];
    // Proceso
    $suma = $n1 + $n2;
?>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    </head>
    <body bgcolor="#D2EBF7">
        <h1>Resultado</h1>

```

```
<table width="168">
  <tr>
    <td width="66"><b>N1</b></td>
    <td width="135"><?php echo($n1) ?></td>
  </tr>
  <tr>
    <td><b>N2</b></td>
    <td><?php echo($n2) ?></td>
  </tr>
  <tr>
    <td><b>Suma</b></td>
    <td><?php echo($suma) ?></td>
  </tr>
</table>
<a href="ejm06.html">Nueva Suma</a>
</body>
</html>
```

El resultado es:

Resultado	
N1	35
N2	76
Suma	111
<a href="#">Nueva Suma</a>	

El método GET también se utiliza con QUERY\_STRING, como ejemplo ilustrativo anote la siguiente URL en el campo **Dirección** del su navegador:

```
http://localhost/php100/Cap04/ejm07.php?N1=78&N2=65
```

Como podemos observar en la misma URL podemos especificar los campos.

## 4 Programas Recursivos

---

En los ejemplos anteriores hemos visto que el formulario está en un documento (\*.html) y el proceso en otro documento (\*.php), también tenemos la opción de hacerlo en un solo documento, donde el formulario envía los datos al mismo documento, el esquema es el siguiente:

```
<?php if(!isset($_POST["control"])){    ?>

    <form method="POST" action="destino.php">

        ----
        ----
        ----

    </form>

<?php } else {

    // Proceso de Datos

} ?>
```

En este caso **control** sería un campo más del formulario, pero oculto para que no se muestre al usuario, y permite verificar si se debe mostrar el formulario ó procesar los datos. También se puede realizar utilizando el método GET, aunque por seguridad se recomienda utilizar el método POST, y para recoger los datos se puede utilizar el arreglo `$_REQUEST` independiente del método de envío (GET ó POST).

### Ejemplo 8

Este ejemplo permite calcular el área de un triángulo.

Archivo: prog2\sem02\ejm08.php

---

```
<?php
$formulario = TRUE;
if( isset( $_REQUEST["control"] ) ){
    // Datos
    $base = $_REQUEST["base"];
    $altura = $_REQUEST["altura"];
    // Proceso
    $area = $base * $altura / 2;
```

```

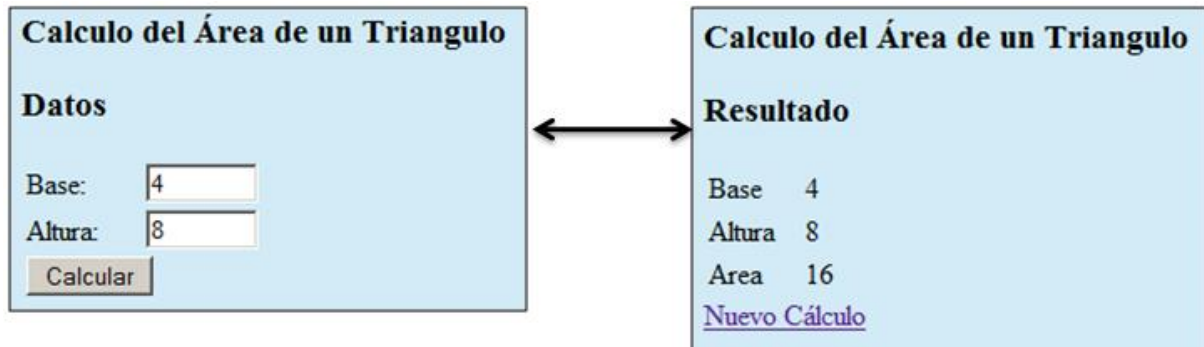
$formulario = FALSE;
}
?>

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>
  <body bgcolor="#D2EBF7">
    <h1>Calculo del Área de un Triangulo</h1>
    <?php if( $formulario ) { ?>
      <h3>Datos</h3>
      <form method="post" action="ejm08.php">
        <input type="hidden" name="control" value="12345">
        <table width="164">
          <tr>
            <td width="62">Base:</td>
            <td width="90">
              <input name="base" type="text" size="6" maxlength="6">
            </td>
          </tr>
          <tr>
            <td>Altura:</td>
            <td><input name="altura" type="text" size="6" maxlength="6"></td>
          </tr>
          <tr>
            <td colspan="2"><input type="submit" value="Calcular"></td>
          </tr>
        </table>
      </form>
      <?php } else { ?>
        <h3>Resultado</h3>
        <table width="138">
          <tr>
            <td width="49">Base</td>
            <td width="77"><?php echo($base) ?></td>
          </tr>
          <tr>
            <td>Altura</td>
            <td><?php echo($altura) ?></td>
          </tr>
          <tr>
            <td>Area</td>
            <td><?php echo($area) ?></td>
          </tr>
        </table>
        <a href="ejm08.php">Nuevo Cálculo</a>
      </?php } else { ?>
    </body>
  </html>

```

```
<?php } ?>  
</body>  
</html>
```

El resultado se ilustra a continuación:



The diagram illustrates the process of calculating the area of a triangle. It consists of two main panels connected by a double-headed arrow, indicating a flow of data from input to output.

**Left Panel: Calculo del Área de un Triangulo**

**Datos**

Base:

Altura:

**Right Panel: Calculo del Área de un Triangulo**

**Resultado**

Base 4  
Altura 8  
Area 16  
[Nuevo Cálculo](#)

## 5 Cookies HTTP

---

PHP soporta cookies HTTP de forma transparente tal y como están definidas en las Netscape's Spec. Las cookies son un mecanismo para almacenar datos en el navegador y así rastrear o identificar a usuarios que vuelven a ingresar a nuestro sitio Web.

Se pueden crear cookies usando la función `SetCookie()`. Las cookies son parte de la cabecera HTTP, así que se debe llamar a la función `SetCookie` antes de que se envíe cualquier salida al navegador. Los datos de una cookie están disponibles en el arreglo con datos de cookies apropiada, tal como `$_COOKIE`.

### Ejemplo 9

Archivo: prog2\sem02\ejm09.php

---

```
<?php
$primera = 0;
if( !isset($_COOKIE["nombre"]) ) {
    setcookie("nombre","Claudia",time()+3600);
    $primera = 1;
}
?>

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body bgcolor="#D2EBF7">
    <?php
    if($primera){
        echo "Hola, esta es tu primera visita.";
    }else{
        echo "Hola " . $_COOKIE["nombre"] . "<br>";
    }
    ?>
</body>
</html>
```



La primera vez que ejecute el programa el resultado es el siguiente:

Hola, esta es tu primera visita.

A partir de la segunda vez, el resultado será este:

Hola Claudia

Para que las cookies funcionen el navegador debe soportarlas y además tienen que estar habilitadas.

## 6 Proyectos Propuestos

---

### 6.1 Proyecto 1

La empresa "EVENTOS PERU" está realizando una serie de presentaciones con artistas nacionales y extranjeros en varios locales de todo el Perú.

La empresa necesita de una aplicación para calcular la rentabilidad del evento, para lo cual se ha considerado lo siguiente:

1. El evento tiene una categoría, que puede ser: LOCAL, INTERNACIONAL o PREMIUM.
2. El local donde se realiza el evento debe tener una capacidad mínima de 15,000 personas, puede ser más.
3. El precio de la entrada al evento y el pago al artista está en función a la categoría del evento según el siguiente cuadro:

CATEGORIA	PRECIO DE ENTRADA	PAGO AL ARTISTA
LOCAL	S/. 10.00	S/. 30,000.00
INTERNACIONAL	S/. 20.00	S/. 70,000.00
PREMIUM	S/. 30.00	S/. 100,000.00

Además se debe considerar los siguientes gastos:

1. Publicidad: 7% del ingreso bruto
2. Logística: 8% del ingreso bruto
3. Transporte: 4% del ingreso bruto (Solo si el evento es en provincia)

Desarrollar la aplicación que requiere la empresa "EVENTOS PERU" para estimar la rentabilidad de un evento. La aplicación debe mostrar un reporte de los INGRESOS, GASTOS y la RENTABILIDAD con sus respectivos detalles.