

Tecnicatura Superior en Análisis de Sistemas

PROGRAMACIÓN V

Tema 5- Formularios, envío de datos y procesamiento.

Para realizar el envío de datos al servidor para luego ser procesados por **PHP**, lo primero que tenemos que hacer es crear un formulario del lado del cliente, es decir en **HTML**. Esto nos va a servir para que el cliente ingrese datos y luego los mande al servidor, el servidor devuelve el resultado de ese envío para ser visualizado por el navegador. En otras palabras, representa una sección de un documento que contiene controles interactivos que permiten a un usuario enviar información a un servidor web.

Para crear un formulario debemos utilizar la etiqueta **form**, seguido de la propiedades **action** y **method**.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Formularios</title>
  </head>
  <body>
    <form action="" method="">

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

El atributo **action** representa la dirección **URL** del archivo que va a procesar los datos, es decir a donde se va a enviar el mismo para luego ser procesado.

```
<form action="mi_archivo.php" method=""></form>
```

Puede ser en el mismo directorio o en otro distinto, pero debemos tener en cuenta de escribir la ruta correctamente, de lo contrario nos daría error. Nosotros vamos a procesar los datos de nuestro formulario en un archivo con extensión **.php** y el mismo puede tener cualquier nombre, siempre es conveniente que tenga un nombre relativo a lo que va a hacer dicho formulario.

Por último tenemos la propiedad **method**, que nos indica la forma en que se enviarán los datos para ser procesados, nosotros veremos dos formas de envío:

- **Post**, los datos del formulario son incluidos en el cuerpo del formulario y son enviados al servidor.
- **Get**, los datos del formulario son adjuntados a la URL del atributo **action** con un signo de pregunta (?) como separador. Y la URL resultante es enviada al servidor. Esta forma de envío es insegura ya que los datos como dije anteriormente, se escriben en la URL, por eso si hacemos un formulario de login por ejemplo, el usuario y contraseña los vería cualquiera sobre la

dirección de la página web de nuestro navegador, y podría ser cambiado fácilmente.



25

En esta imagen vemos como una multiplicación que realizo el usuario en dos text llamados **text1** y **text2** es enviada al servidor por medio de la URL utilizando **GET**. También podemos observar el valor que ingreso el usuario (**5**) en cada uno de los text, como así también que fue lo que eligió de las opciones de la lista (**multiplicar**). En el caso que quisiéramos cambiar un valor que envió el usuario, simplemente haciendo clic en la **URL** del navegador por ejemplo, en el **text1=5**, lo cambiamos por 7 y refrescamos (F5), veríamos otro resultado (35). Sin que tengamos que cambiar algo en el archivo .php del servidor o en el archivo HTML

En resumen un formulario básico en HTML tiene la siguiente estructura:

```
<form action="archivo.php" method="POST">
  <!--Contenido del formulario  -->
  <label for="nombre">Ingrese su nombre:
</label>
  <input type="text" name="nombre"
placeholder="ingrese su nombre">

  <!--Botón para el envío de datos  -->
  <input type="submit" value="Enviar">
</form>
```

En este ejemplo, el formulario cuenta con un control HTML **input** del tipo **text**, en donde el usuario debe ingresar su nombre y después hacer clic en el botón **Enviar**, para mandar esos datos que ingresó al servidor para ser procesados por medio del lenguaje de programación PHP.

En la siguiente imagen vemos el resultado desde el navegador, más adelante veremos cómo darle forma utilizando estilos CSS y Bootstrap:

Ingrese su nombre:

Controles de un formulario (HTML 4 y HTML5)

Un formulario contiene –además de los elementos típicos, como párrafos, imágenes y listas– un conjunto de **controles**, llamados también **componentes** o **campos de formulario**, que le van a permitir al usuario ingresar, seleccionar y visualizar información.

Etiqueta (Label)

El control **label** (**etiqueta**) se emplea en un formulario para mostrar información que el usuario no puede modificar, como por ejemplo descripciones, mensajes de ayuda, y para identificar a otros controles, como por ejemplo los cuadros de texto.

```
<form action="miArchivo.php" method="post">  
  
    <label>Esta es una etiqueta o label </label>  
    <input type="text" name="texto" >  
  
</form>
```

TÍTULO

Mi formulario
Esta es una etiqueta o label

PIE DE PÁGINA

La etiqueta **label** tiene el atributo **for**. Este atributo es opcional; cuando se lo agregamos a un control **label**, estamos indicando el identificador (con el atributo **id**) al cual pertenece el título.

```
<label for="nombre">Nombre</label> <br/>  
<input type="text" id="nombre" name="nombre" />
```

En el ejemplo anterior, podemos observar cómo le agregamos el atributo **for** al **label** “**nombre**” y, a continuación, agregamos un input del tipo **text** con el atributo **id=“nombre”**. De esta forma, estamos indicando que el **label** “**nombre**” pertenece al **text** con el identificador “**nombre**”.

Otra opción interesante que cumple la misma función de un **label** dentro de un formulario es el atributo **placeholder**.

El **placeholder** representa un texto corto que muestra una sugerencia o ayuda para que el usuario ingrese información en un cuadro de texto o en un cuadro de búsqueda. El texto que muestra este atributo se visualiza como una marca de agua dentro del control en el que se aplicó el atributo.

A diferencia de las páginas web convencionales, este atributo se utiliza mucho en dispositivos móviles y siempre tiene que ir sobre el componente que queremos describir.

En el siguiente ejemplo, agregamos el atributo **placeholder** a un cuadro de texto:

```
<form action="miArchivo.php" method="post">

    <input type="text" name="texto"
    placeholder="Ingrese su nombre">

</form>
```

TÍTULO

Mi formulario

PIE DE PÁGINA

Campo de texto (Text input)

Los **campos de texto**, **text input**, o, simplemente, **campos de entrada**, se utilizan para recibir información generada por el usuario.

Como mencionamos anteriormente, si solamente vamos a mostrar o visualizar texto usamos el control **label**, pero si, en cambio, necesitamos que el usuario ingrese algún dato, debemos usar los campos de texto.

Tipos de text input

Existen distintos tipos de text input en HTML5 como la contraseña, teléfono, e-mail y varios más. Solamente tenemos que agregar al atributo **type** el valor que necesitemos.

Los valores que podemos asignarle al atributo **type** son los siguientes:

text, **password**, **email**, **tel**, **url**, **search**, **number**, **textarea**, **time**, **date**, **week**, **datetime**, **color**, **range** y **submit**.

```
<form action="index2.php" method="post">
  <label>Contraseña</label>
  <input type="password" name="pass">
  <label>Numero</label>
  <input type="number" name="numero">
  <label>E-mail</label>
  <input type="email" name="email">
  <label>Url</label>
  <input type="url" name="url">
  <label>Teléfono</label>
  <input type="tel" name="tel">
  <label>Hora</label>
  <input type="time" name="hora">
  <label>Fecha</label>
  <input type="date" name="fecha">
  <label>Día</label>
  <input type="date" name="dia">
  <label>Semana</label>
  <input type="Week" name="semana">
  <label>Fecha y hora</label>
  <input type="datetime" name="fechayhora">
  <label>Color</label>
  <input type="color" name="color">
  <input type="submit" value="enviar">
</form>
```

título

Mi formulario

Contraseña

Numero

1

E-mail

mi@mail.com

Url

http://www.redusers.com/noticias/

Telefono

1566669999

Hora

23:15

Fecha

13/09/2014

Dia

07/11/2014

Semana

Semana 24, 2014

Fecha y hora

07/11/2014 23:15:52

Color

enviar

PIE DE PÁGINA

El control tipo **text** nos permite generar un cuadro de texto de una sola línea. Sus parámetros más importantes son: **name** que permite establecer el nombre con el que haremos referencia al cuadro de texto en el código y **maxlength** que especifica la longitud máxima de caracteres permitidos para el cuadro de texto. Si no se establece este parámetro, se permite cualquier longitud de texto.

```
<label for="nombre">Ingrese su nombre:</label>
<input type="text" name="nombre" maxlength= "25">
```

El control tipo **password** ('contraseña'), es igual al cuadro de texto y se genera agregando el valor **password** al atributo **type**. Los valores que ingresemos en el cuadro de texto se verán con puntos o asteriscos para que no se pueda visualizar el texto escrito en el mismo. Tengamos en cuenta que, en algunos dispositivos, el carácter que se ingresa se muestra durante un segundo antes de que sea cambiado por el carácter asterisco o punto.

```
<label>Ingrese su contraseña</label>
<input type="password" name="contraseña">
```

En los casos de los tipos **números** (`type="number"`) o **número de teléfono** (`type="tel"`), el framework interpreta que se trata de números, entonces nos facilita el ingreso, mostrándonos el teclado numérico y no el estándar.

En el caso del control de tipo número, aparte de mostrarnos un teclado simplificado, si tocamos sobre el campo de texto, se despliega una lista ordenada con los números negativos y positivos, para poder seleccionar el que necesitemos sin que tengamos que tipearlo sobre el teclado.

Con los controles del tipo **email** y **url** pasa exactamente lo mismo: el dispositivo nos muestra un teclado con **@** y **.com** y otros caracteres especiales, para facilitarnos el ingreso de datos. Además al definirlo como tipo **email** o **url**, estamos restringiendo el ingreso de datos a un formato válido.

Los controles que son del tipo **fecha** (`type="date"`), **fecha y hora** (`type="datetime"`) y **semana** (`type="week"`) vienen predeterminados directamente, es decir, si tocamos sobre el componente lista se despliega un calendario para poder seleccionar un día, fecha o semana, que necesitemos agregar al cuadro de texto. En el caso de un control del tipo **hora** (`type="time"`), al tocar el campo cuadro de texto podemos visualizar el formato de **hh:mm** para poder escribir sobre él, o bien seleccionar en la lista desplegable el número que necesitemos usar.

Para generar un control que nos permita seleccionar colores de una lista, usamos el tipo **color** (`type="color"`).

El tipo **submit** (`type="submit"`) nos crea un botón de envío. Al pulsarlo, todos los datos ingresados en los controles del formulario se enviarán para ser procesados.

Textarea

Si bien el text input y el textarea nos sirven para el ingreso de datos por parte del usuario, la diferencia que existe entre ambos es que **textarea** permite el ingreso de **texto multilínea** en el formulario.

Si tan sólo colocamos el atributo `<textarea>`, va a permitir al usuario ingresar más de una línea de texto.

Un ejemplo clásico es un formulario de consulta, en donde el usuario ingresa en el textarea la consulta y esta es enviada al servidor para luego ser procesada como se muestra en el siguiente ejemplo:

```
<form action="index2.php" method="post" >
    <label >Consulta:</label>
    <textarea name="textarea"
id="textarea"></textarea>
    <input type="submit" value="enviar" >
</form>
```

The screenshot shows a web browser window with a form titled "Mi formulario". Inside the form, there is a label "Consulta:" followed by a text area. The text area has a placeholder text "Esto es una consulta...". Below the text area is a button labeled "enviar". The form is set against a light gray background with a header "TITULO" and a footer "PIE DE PÁGINA".

Campo de búsqueda (Search input)

También disponemos, en el atributo **type**, del valor **search** que nos sirve para ingresar valores de búsqueda. Este cuenta con un icono –ya predefinido– de una lupa en el costado izquierdo. Cuando ingresamos texto, en el costado derecho del cuadro de texto aparece un icono en forma de cruz, que nos permite al presionarlo borrar lo que escribimos.

Podemos usar la siguiente sintaxis para crear un campo de búsqueda:

```
<form action="index2.php" method="post" >
  <label>Buscar:</label>
  <input type="search" name="buscar" id="buscar">
</form>
```



The screenshot shows a web form with a title bar labeled 'TÍTULO'. Below the title bar is a section labeled 'Mi formulario'. Inside this section, there is a label 'Buscar:' followed by a search input field. The input field has a magnifying glass icon on the left and a clear icon on the right. The placeholder text inside the input field is 'Parámetro a buscar...'. Below the input field is a button labeled 'enviar'. At the bottom of the form, there is a footer bar labeled 'PIE DE PÁGINA'.

Slider

El **slider** es otro valor que recibe el atributo **type** de la etiqueta **input** (**<input type="range">**). Los sliders nos permiten ajustar un valor entre un rango determinado mediante un componente deslizable.

El slider viene acompañado de un text input a la derecha, en el cual podemos ingresar un valor que será asociado directamente a la barra deslizante. Lo mismo ocurre si nos deslizamos con la barra: nos mostrara el mismo valor en el text input de la derecha, por lo que siempre estarán sincronizados asegurándonos que tengan el mismo valor para luego ser procesados.

Los atributos que tenemos que tener en cuenta para configurar el slider son el valor de comienzo del rango, el valor final y el valor de incremento.

Para esto disponemos de los atributos **min** y **max** que serían, respectivamente, el valor inicial y el final. Para el incremento, contamos con el atributo **step**.

Para darle un mejor aspecto contamos también con el atributo **data-highlight** que, pasándole por valor **"true"** nos pinta en color el valor utilizado en la barra.

La sintaxis para crear un control del tipo slider es la siguiente:

```
<label>Slider: </label>
<input type="range" name="slider" id="slider"
max="100" min="00" value="25" step="20" data-
highlight="true" />
```


Mi formulario

Slider:

40

Botón de opción (radio button)

El control **radio button** ('**botón de opción**') permite, tener más opciones: no solamente tendríamos verdadero o falso, sino que nos sirve cuando necesitamos que el usuario elija una **sola opción** entre varias.

Un ejemplo podría ser la elección de pago de un determinado producto, donde las opciones a elegir por el usuario serían: "**Pago en efectivo**", "**Pago con tarjeta de crédito**" o "**Pago con tarjeta de débito**". En este ejemplo, necesitamos una sola opción para después ser procesada. La elección del usuario nos devolvería solamente un valor, pero con este control podría elegir entre varias.

La sintaxis que nos permite incluir este tipo de controles en un formulario es la siguiente:

```
<fieldset>
  <legend>Titulo del que se visualiza con el
grupo:</legend>
  <input name="radio" id="radio-1" type="radio"
value="opción1" checked>
  <label for="radio-1">Opción 1</label>
  <input name="radio" id="radio-2" type="radio"
value="opción2">
  <label for="radio-2">Opción 2</label>
</fieldset>
```

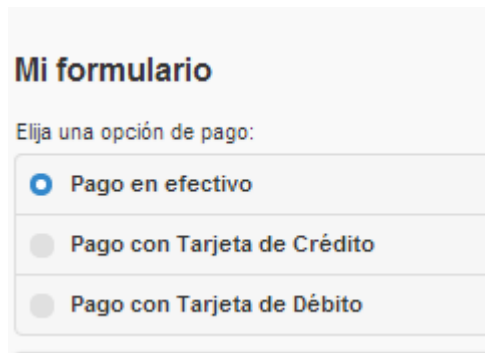
Como podemos observar en el código anterior, para incluir un radio button en un formulario tenemos que usar primero la etiqueta **<fieldset></fieldset>** que se usa en HTML para agrupar campos relacionados, seguido, de manera opcional, por una etiqueta **<legend></legend>**, en el caso de que quisiéramos ponerle un título al grupo de opciones.

Luego a la etiqueta **<input></input>** al atributo **type** se le asigna el valor **radio**. Es importante que a continuación agreguemos las etiquetas **<label></label>** con el atributo **for** que deberá ser igual al **id** correspondiente del radio button.

En el ejemplo que detallamos a continuación, se le solicita al usuario que elija una opción de pago:

```
<fieldset >
  <legend>Elija una opción de pago:</legend>
```

```
<input name="radio" id="radio-1" type="radio"
value="efectivo" checked>
<label for="radio-1">Pago en efectivo</label>
<input name="radio" id="radio-2" type="radio"
value="credito">
<label for="radio-2">Pago con Tarjeta de
Crédito</label>
<input name="radio" id="radio-3" type="radio"
value="debito">
<label for="radio-3">Pago con Tarjeta de
Débito</label>
</fieldset>
```



The screenshot shows a web form with the title "Mi formulario". Below the title is a label "Elija una opción de pago:". There are three radio button options listed below: "Pago en efectivo" (which is selected, indicated by a blue dot), "Pago con Tarjeta de Crédito", and "Pago con Tarjeta de Débito".

Casilla de verificación (checkbox)

El control **checkbox** ('Casilla de verificación') se utiliza para permitirle al usuario seleccionar una condición sí o no, verdadero o falso; este control, a diferencia del radio button, nos permite hacer múltiples selecciones.

Podemos incluir la cantidad que nuestra aplicación necesite. Por ejemplo, podemos preguntarle al usuario que tipo de deporte que práctica, y no necesariamente puede elegir una sino que puede elegir varias opciones.

Para agregar un solo checkbox al formulario la sintaxis para hacerlo sería la siguiente:

```
<label><input type="checkbox" name="checkbox-0" />
Acepto el contrato</label>
```

En el código anterior, se le solicita al usuario que acepte el contrato. Este ejemplo es muy usado para aceptar términos y condiciones en sitios web.

El siguiente ejemplo nos muestra cómo establecer un grupo de opciones dentro de un formulario, para permitir que el usuario nos indique cuáles son los deportes que practica.

```

<fieldset>
  <legend>Seleccione el o los deportes que practica:</legend>
  <input name="checkbox-1" id="checkbox-1" type="checkbox">
  <label for="checkbox-1">Fútbol</label>
  <input name="checkbox-2" id="checkbox-2" type="checkbox">
  <label for="checkbox-2">Tenis</label>
  <input name="checkbox-3" id="checkbox-3" type="checkbox">
  <label for="checkbox-3">Béisbol</label>
  <input name="checkbox-4" id="checkbox-4" type="checkbox">
  <label for="checkbox-4">Natación</label>
</fieldset>

```

Mi formulario

Seleccione el o los deportes que practica:

<input type="checkbox"/>	Fútbol
<input checked="" type="checkbox"/>	Tenis
<input type="checkbox"/>	Beisbol
<input checked="" type="checkbox"/>	Natación

Selector de fechas (datepicker)

El **datepicker** (**selector de fechas**), es un componente que nos permite seleccionar una fecha dentro de un calendario. Para hacerlo debemos escribir el siguiente código:

```

<input type="date" min="2010-01-01" value="2014-07-18">

```

Calendario:

July 2014						
Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

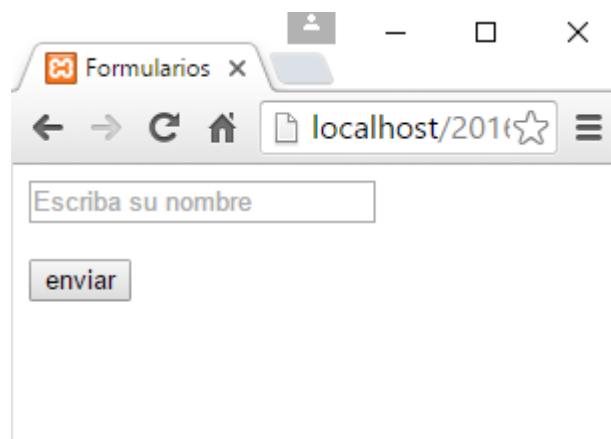
Botón submit

Por ultimo nos queda saber de qué manera el usuario va a enviar la información, para esto al finalizar utilizaremos un botón llamado **submit**.

```
<input type="submit" value="Enviar información">
```

Este botón es un componente **input** que por propiedad **type** recibe el valor **submit**.

Lo que nos genera el código anterior, es un botón clásico de enviar, que al ser presionado por el usuario enviará los datos al servidor para luego ser procesados a la URL que indicamos en la propiedad **action** del formulario.



The screenshot shows a web browser window with a single tab titled 'Formularios'. The address bar displays 'localhost/2016'. The page content includes a text input field with the placeholder text 'Escriba su nombre' and a button labeled 'enviar' below it.

Para tener en cuenta

Es importante al confeccionar un formulario tener en cuenta la propiedad **name** del componente, ya que con ese mismo valor, va a ser llamado desde el archivo que

procesara los datos en el servidor. Y si, además le agregamos un identificador (id) o una clase (class), ese mismo nombre también utilizaremos para hacer referencia al mismo desde la hoja de estilo CSS. **Por eso es importante escribir bien los nombres, id o clases de todos estos elementos HTML, respetando mayúsculas y minúsculas, y que siempre, el mismo, tenga alguna referencia a lo que hace.** Como por ejemplo si en un ingreso de texto, el usuario va a ingresar el e-mail, podemos llamarlo **txt_mail**, así de esta forma, nos facilitaría mucho el procesamiento de estos datos desde **PHP**.

Estos son algunos de los controles de un formulario, los que más se utilizan o utilizaremos en este curso. Les recomiendo la página del consorcio W3C donde encontrarán más información: http://www.w3schools.com/html/html_forms.asp

Nota: La mayoría de la información de formularios **HTML** fue extraída del libro *jQuery Mobile* del editorial Users, que es de mi autoría.

Les dejo también un video para ampliar conocimientos: <https://www.youtube.com/watch?v=QDOdODwsfuQ>

Otra web con controles: https://www.w3.org/community/webed/wiki/Es/Elementos_de_formulario_adicionales_d_el_HTML5

Procesamiento de datos a través del servidor

Una vez que tenemos en claro la parte del lado del cliente, es decir cómo se debería crear un formulario para ser enviado; nos queda saber cómo va a ser procesado por el servidor y de qué manera nos va a devolver la información, si es que nos devuelve algo...

Para entenderlo, veremos un ejemplo práctico:

Del lado del cliente quiero que el usuario ingrese su nombre, el mismo sea procesado y me devuelva el siguiente mensaje: "Hola **[Nombre del usuario]**, bienvenido a la web".

Para el formulario HTML necesitaremos un **label**, y un input del tipo **text**, más el botón **submit** (este archivo podría llamarse index.html o como Uds. prefieran):

```
<form action="procesar.php" method="POST">
  <label for="txt_nombre">Escriba su nombre:
</label>
  <input type="text" name="txt_nombre">
  <input type="submit" value="Enviar">
</form>
```

Escriba su nombre:

El siguiente paso es crear el archivo **PHP** que va a procesar los datos, a dicho archivo lo llamaremos **procesar.php**. Recordemos que el nombre debe ser el mismo que pusimos en el **action** del formulario.

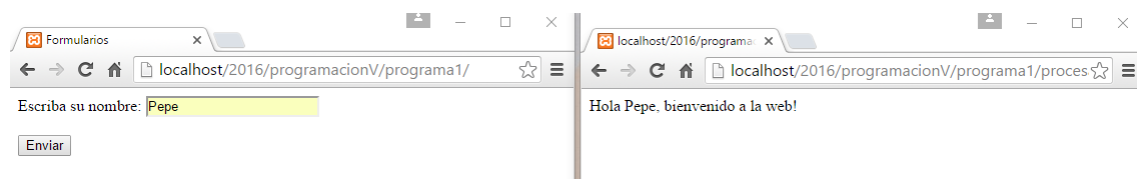
```
<?php

    $nombre = $_POST['txt_nombre'];

    echo "Hola ".$nombre.", bienvenido a la web!";

?>
```

Y el resultado sería el siguiente:



La forma en PHP de recoger el **elemento** enviado desde HTML es con `$_POST['NAME DEL INPUT']`, que en el ejemplo anterior se lo estoy pasando a una variable llamada **\$nombre**. Por ultimo con el comando **echo** muestro al usuario la frase pedida, concatenando con la variable.

Entonces `$_POST ['NOMBRE DEL ELEMENTO']` o bien `$_GET[' NOMBRE DEL ELEMENTO ']`, nos sirve para recoger el valor enviado por el formulario **HTML** (depende siempre el método de envío, si utilizo GET recibo por `$_GET` y si utilizo POST recibo con `$_POST`); y puedo pasarlo a una variable o utilizarlo directamente.

Si tuviera que sumar dos valores podríamos hacer lo siguiente:

En el archivo **HTML** que podemos llamar también **index.html**, agregamos un formulario y en su interior dos **labels**, dos **texts** y un botón **submit**.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Suma</title>
</head>
<body>
    <form action="procesar.php" method="POST">
        <label for="txt_n1">Ingrese Número 1:
</label>
```

```

        <input type="text" name="txt_n1"><br>
        <label for="txt_n2">Ingrese Número 2:
</label>
        <input type="text" name="txt_n2"><br><br>
        <input type="submit" value="Sumar">
    </form>
</body>
</html>

```

Nota: utilizo **<meta charset="UTF-8">** para que se puedan ver los acentos en la página, caso contrario no se verían (más información sobre codificación en: <https://es.wikipedia.org/wiki/UTF-8>).

Ingrese Número 1:

Ingrese Número 2:

En el archivo **PHP** que llamaremos **procesar.php** hacemos lo siguiente:

```

<?php

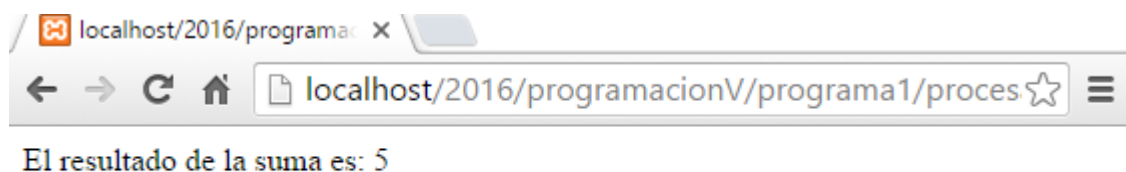
    $resultado = $_POST['txt_n1'] +
    $_POST['txt_n2'];

    echo "El resultado de la suma es:
    ".$resultado;

?>

```

En la variable **\$resultado** guardo el resultado de la suma de los dos valores recibidos por **POST** (txt_n1 y txt_n2). Por último muestro el resultado concatenado con un texto utilizando el comando **echo**.



Actividades

1. Realizar un formulario HTML que tenga las cuatro operaciones básicas (suma, resta, multiplicación y división). Y un archivo PHP que procese la información enviada por la página HTML, mostrando el resultado de la operación elegida. En el diseño pueden utilizar un combo para seleccionar la operación deseada:

```
<select name="lista" >
  <option value="sumar">Sumar</option>
  <option value="restar">Restar</option>
  <option
value="multiplicar">Multiplicar</option>
  <option value="dividir">Dividir</option>
</select>
```

2. Realizar una página web que pida información al usuario, nombre, apellido, teléfono y email. Dicha información deberá ser procesada y al ejecutar el botón submit, le tiene que mostrar al usuario la siguiente frase: "*Bienvenido al sistema [Nombre][Apellido], su teléfono es: [Telefono] y su e-mail es: [Mail]*".
3. Realizar un sitio web en donde le pida al usuario su **Nombre** y su **edad**. La misma deberá ser procesada y si el usuario ingreso que su edad es mayor a 18 años, mostrarle el siguiente mensaje: "*Bienvenido [Nombre] al sistema*", en caso de que sea menor a 18 años el mensaje será el siguiente: "*[Nombre] Ud, no puede ingresar al sistema*".
4. Realizar una página web que pida al usuario tres lados de un triángulo, procese los datos y muestre el resultado al usuario diciéndole que tipo de triángulo es.
5. Al formulario del punto 3 agregarle dos radio buttons que le digan al usuario si acepta los términos y condiciones para entrar a la web, de aceptarlos y tener la edad mayor a 18 años mostrarle el mensaje "*Bienvenido [Nombre] al sistema*", en caso de que el usuario sea menor a 18 años o no acepte los términos y condiciones para entrar a la web le diga: "*[Nombre] Ud., no puede ingresar al sistema*".

Ingrese su Nombre:

Ingrese su edad:

Términos y condiciones

☐ Acepto ☒ No acepto

Autoevaluación

1. ¿Para qué sirven los formularios?
2. ¿Qué hace la propiedad **method** de un formulario?
3. ¿Qué hace la propiedad **action** de un formulario?
4. ¿Qué diferencia hay entre los valores **GET** y **POST**?
5. ¿Qué tipo de valor debe recibir la propiedad **type** de un elemento **input** para enviar los datos a un archivo **PHP**?