

Gestión de Formularios

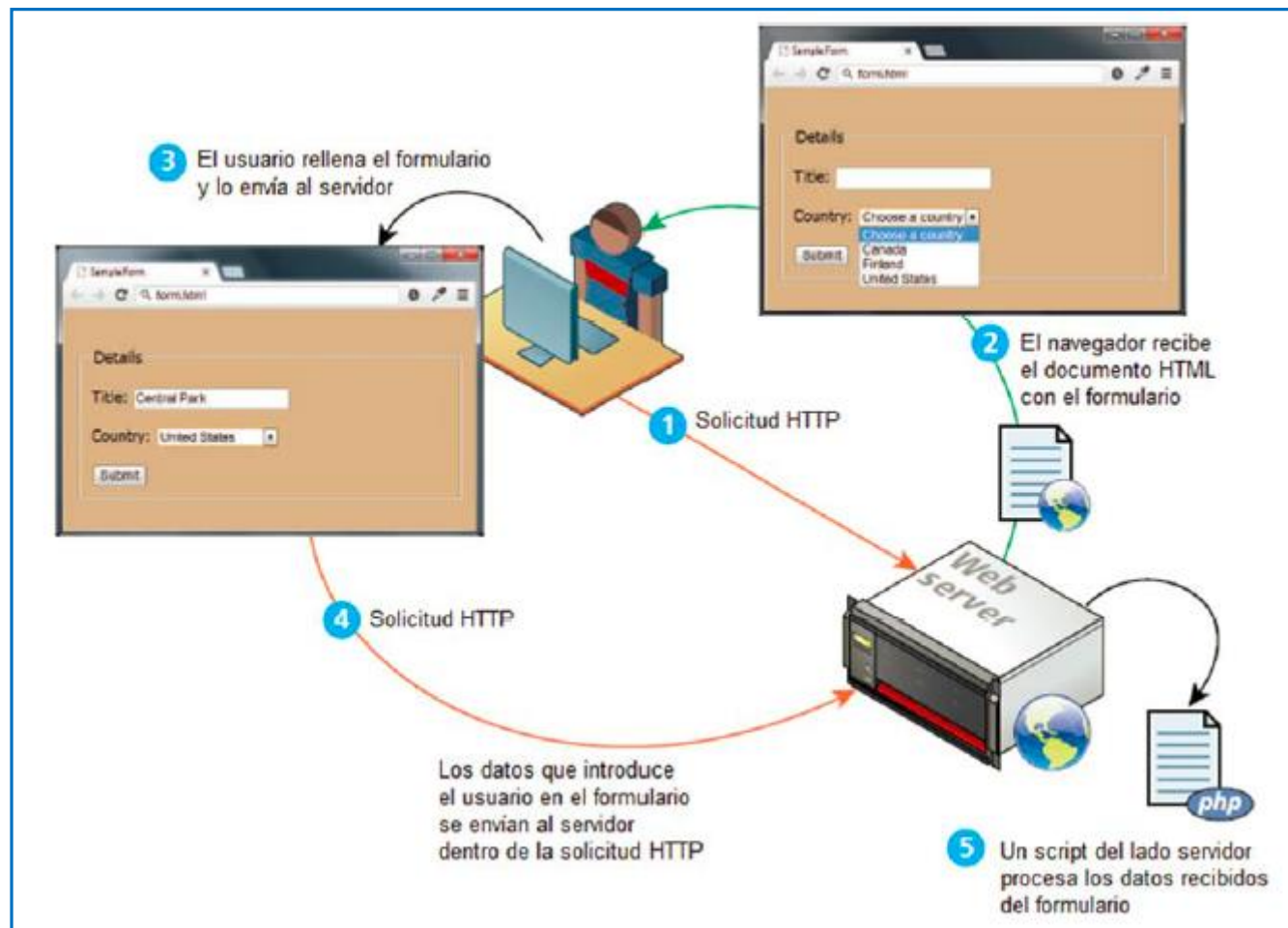
- **Formularios:**

*Elemento fundamental en las páginas web para **recopilar datos** de los usuarios, **enviando** dicha información a un **servidor web** para que la **procese***

- Imprescindibles para obtener información de los usuarios y adaptar los contenidos de las páginas web a sus necesidades.
 - Ej: solicitud de información, registro de usuarios, encuestas, buscador, compra de un producto, etc.
- La **interactuación** entre una página web y los usuarios **permite**:
 - 1) **Recabar datos** de los usuarios.
 - 2) **Enviar** dicha información a un **servidor** (web/de aplicaciones) indicando su dirección URL.
 - 3) El **servidor**, a través de un **script**, **procesa** los datos y envía un **mensaje respuesta al usuario**.
- Las **páginas web** se convierten en **dinámicas** gracias a los formularios.

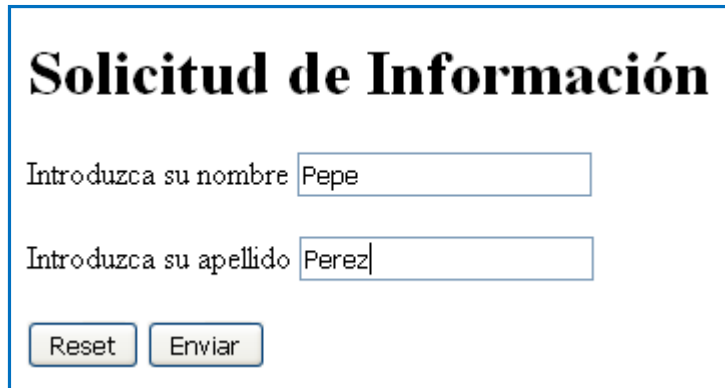
Formularios

- Permiten enviar datos al servidor para:
 - ✓ Interactuar con una **bd**
 - ✓ Producir una **salida** en el **navegador**



- **Componentes software** que se utilizan con los **formularios**:
 - ✓ **Página que contiene el formulario: código HTML** para mostrar el formulario (*interfaz* de usuario de la aplicación).

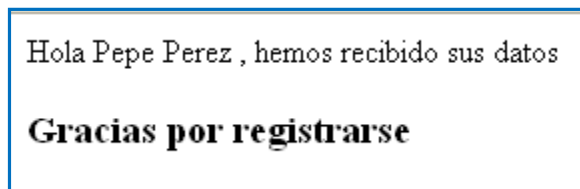
- Ej:



The screenshot shows a web form titled "Solicitud de Información". It contains two input fields: "Introduzca su nombre" with the value "Pepe" and "Introduzca su apellido" with the value "Perez". Below the input fields are two buttons: "Reset" and "Enviar".

- ✓ **Procesamiento en el servidor: código PHP** (u otro lenguaje de script del lado de servidor) para **procesar** los datos recogidos.
 - Lo habitual es que esté en un fichero **diferente** a la página que muestra el formulario.

- Ej:



The screenshot shows a confirmation message: "Hola Pepe Perez , hemos recibido sus datos" followed by "Gracias por registrarse" in bold.

Formularios

- Los **datos** introducidos por el usuario a través de los controles se almacenan en una **variable superglobal** al enviar el formulario a través del botón **submit**.
 - El **dato** que envía **cada control** se almacena en un elemento del array asociativo: **\$_GET**, **\$_POST** o **\$_REQUEST**.
 - El nombre del control -valor del atributo **name**- da lugar a la **clave** del elemento.
 - Si se usa el **método get**, los datos se guardan en el superarray **\$_GET**.
 - Si se usa el **método post**, los datos se recogen en el superarray **\$_POST**.
- **Ej1:** crea una página html que contenga un formulario, en el que se solicita el nombre del usuario. Gestiona dicha información a través del método “**get**”.
- **Pista:** `$nombre=$_GET[“nombre”]`

Solicitud de Información

Introduzca su nombre

Introduzca su apellido

Los datos de la super-global `$_GET` son:

```
Array
(
    [Nombre] => Juanma
    [Apellido] => Rodriguez
)
```

Hola Juanma Rodriguez , hemos recibido sus datos

Gracias por registrarse

Formularios

- **Sol. Ej1:** desde la barra de dirección del navegador se tienen que ver los datos que se envían al servidor.

```
localhost/12_ej1_a_form_get.php?nombre=Juanma&apellido=Rodríguez
```

```
<?php
    $varNom = $_GET["Nombre"];
    $varApel = $_GET["Apellido"];

    echo "Los datos de la super-global \$_GET son: <pre>";
    print_r($_GET);
    echo "</pre>";

    echo "<br /> Hola " . $varNom . " " . $varApel . " , hemos recibido
    sus datos <br />
    <h3>Gracias por registrarse</h3>";
?>
```

Validación de Datos

▪ Validación de datos:

- ✓ Es necesario **verificar** que se han cubierto todos los datos **obligatorios**, que éstos son **válidos** y que son **seguros**.
- ✓ Si los datos no están completos o **no son válidos**, hay que **notificar al usuario**.
- ✓ Si algún dato **no es seguro**, hay que **rechazarlo** (*saneado de entradas*).
- ✓ Todos los controles y áreas de texto **cubiertos crean elementos** en el **array asociado**.
- ✓ Hay que **eliminar espacios innecesarios** y **código malintencionado**, comprobar si el **tipo de datos** es el solicitado y si los datos están dentro del **rango** permitido.
- ✓ Sólo los *botones de radio* y las *casillas de verificación* marcados **crean elementos** en el **array** asociado.

▪ ¿Dónde se validan los datos?

a) En el **lado cliente**:

- ✓ con **atributos** de **HTML5** –*required, maxlength, ...*–
- ✓ con un lenguaje de script –*Javascript*–

b) En el **lado servidor**:

- ✓ usando **funciones internas de PHP** –*is_numeric()*, ...–

Validación de Cuadros de texto

- **Cuadros de texto:**
- El usuario puede escribir cualquier **texto** o **número**.
- Los **nº** se **interpretan** como **texto**.
- **Sintaxis:** `<input type= "text" name="valor">`
- **Ej:** `<input type="text" name="nombre" maxlength="20" required>`
- ❖ **Validaciones habituales:**
 - ✓ Saber si una variable está **definida**.
 - ✓ Saber si tiene algún dato **cubierto**.
 - ✓ Saber si es **numérica** o de cualquier otro tipo.
 - ✓ Limitar el **tamaño** máximo de un string.
 - ✓ Eliminar cifras decimales.
 - ✓ **Eliminar espacios en blanco** innecesarios y redundantes.
 - ✓ Ignorar etiquetas de Html y **eliminar código malintencionado**.
- Para realizar las validaciones anteriores, se utilizan **funciones internas** de PHP.

Validación de Cuadros de texto

❖ Funciones de validación:

- ✓ **empty()**: determina si una variable está vacía -0, ""- o no existe
- ✓ **is_numeric()**: determina si una variable tiene un valor numérico
- ✓ **is_float()**: determina si una variable es decimal
- ✓ **is_string()**: determina si una variable contiene una cadena de caracteres
- ✓ **is_array()**: determina si una variable es un array
- ✓ **strlen()**: devuelve el tamaño en caracteres de un string
- ✓ **round()**: redondea un número decimal. Por defecto, devuelve el número sin ninguna cifra decimal
- ✓ **trim()**: elimina espacios en blanco del inicio y del final de una cadena de caracteres. Devuelve la cadena recortada. También: **ltrim()** y **rtrim()**
- ✓ **stripslashes()**: elimina caracteres \
- ✓ **strip_tags()**: retira las etiquetas HTML y PHP contenidas en una cadena de caracteres de los controles de un formulario

Validación de Cuadros de texto

- **Ej2:** crea una página html que contenga un formulario, en el que se solicita el nombre y los apellidos de los clientes. Verifica que no contiene ningún carácter de Html ni de Php.

Solicitud de Información

Datos personales

Introduzca su nombre:

Introduzca sus apellidos:

Los datos de la super-global \$_GET son:

```
Array
(
    [nombre] => Maria
    [apellidos] => Rodríguez López
)
```

El nombre a insertar en la bd es: Maria

Los apellidos son: **Rodríguez López**

localhost/12_ej2_a_input_text_valida.php?nombre= Maria<%2Fb>&apellidos= Rodríguez+López<%2Fb>

Validación de Cuadros de texto

➤ **Ej3:** crea una página html que contenga un formulario que pida qué **artículo** se quiere comprar y cuántas **unidades**.

■ Requisitos:

- ✓ Verifica que se han cubierto ambos campos (campos obligatorios).
- ✓ Elimina posibles espacios en blanco.
- ✓ El nº de unidades debe ser un valor numérico.
- ✓ No sirven los nº negativos ni los nº decimales.
- ✓ La cantidad máxima a indicar es de 999.

Artículo: Unidades:

Artículo: Unidades:

Artículo: ipod
Unidades: 3

Artículo: Unidades:

Unidades:

Error, introduzca todos los datos, por favor

Artículo: ipod
Error, las unidades tiene que ser un numero mayor o igual que 1 y menor de 999

Validación de Cuadros de texto

➤ Variaciones sobre el Ej3:

- a) Comprobar de **forma individual** si cada campo está cubierto y mostrar **diferentes mensajes** de error, según si es un control u otro el que está vacío. Para el campo unidades, mostrar el resto de errores.

Artículo: Unidades:

Artículo: naranjas
Unidades: 5

Artículo: naranjas
Error, introduzca el numero de unidades

Error, introduzca el articulo deseado
Error, introduzca el numero de unidades

Error, introduzca el articulo deseado
Error, las unidades tiene que ser un numero mayor o igual que 0 y menor de 999

Validación de Contraseñas

■ Contraseñas:

- El usuario puede escribir cualquier **texto** o **número**.
- El valor introducido no se muestra en pantalla (círculo o asterisco)..
- **Sintaxis:** `<input type= "password" name="contraseña">`
- **Ej:** `<input type="password" name="contra" maxlength="15" required>`

❖ Validaciones habituales:

- ✓ Saber si se ha **cubierto**.
 - ✓ Exigir un **tamaño mínimo** de la contraseña.
 - ✓ Limitar el **tamaño máximo** de la contraseña.
- **Ej4:** crea una página html que contenga un formulario que solicite la **contraseña**.
- **Requisitos:**
 - ✓ Verifica que se ha cubierto la contraseña (campo obligatorio).
 - ✓ Comprueba que su tamaño está entre 6 y 10.

Contraseña:

Error: la contrasena debe tener un maximo de 10 caracteres.

Error: la contraseña debe tener al menos 6 caracteres.

Validación de Cuadros de texto

- **Ej5:** Crea un formulario que solicite el **nombre** y la **nota** obtenida en un examen. Una vez recogidos esos datos, se mostrará en pantalla dicho nombre, la nota, la evaluación correspondiente y una serie de observaciones que dependerán de la nota.
- **Requisitos:**
 - ✓ Verifica que se han cubierto ambos campos (campos obligatorios).
 - ✓ Las notas tienen que estar comprendidas entre 0 y 10.
 - ✓ Los nº flotantes se redondean.

Gestión de notas

Por favor, introduzca correctamente los siguientes datos

Alumno:

Nota:

Reset

Enviar

Hola "pepe", tu evaluacion es de "Notable" . Muy bien, pasas de curso con una buena nota

Por favor, rellene los datos.

Error, la nota tiene que ser un número entre 0 y 10

Validación de Datos en la misma página

Envío de datos a la misma página: útil cuando queremos **validar** la información y **mostrar** en el **mismo fichero** qué valores no se han cubierto correctamente.

Mediante la variable **superglobal** **\$_SERVER**: consultar dinámicamente el valor de la **clave** **PHP_SELF**, que devuelve el nombre del script que se está ejecutando actualmente. También incluye la ruta. Es la **mejor forma** de averiguar el valor del nombre del fichero actual.

- **Ej:** `<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">`

Comprobar método de envío: útil para saber si el formulario se ha enviado utilizando el **método** *get* o *post*. **Formas de hacerlo:**

a) Utilizando el array global **\$_SERVER** con la **clave** **REQUEST_METHOD**:

- **Ej:** `if ($_SERVER['REQUEST_METHOD'] == 'GET') {...}`

b) Dándole un nombre interno **-name-** al botón del **submit** y preguntar si ese valor está definido en el array correspondiente (**\$_GET** o **\$_POST**).

- Es la opción **recomendable** si hay **varios formularios** en la misma página.
- **Ej:** `if (isset($_GET['submit'])) {...}`

Sintaxis alternativa de Sentencias de Control

➤ **Sol:** utilizando dos bucles **for anidados** con su **sintaxis alternativa**

- **Sintaxis:** `<?php for ($i=valor; $i<condicion; $i++) : ?>`

código Html *//etiqueta de apertura en Html*

código Php

código Html

código Php

.....

`<?php endfor; ?>`

```
<html>
  <h2>Sintaxis alternativa de las sentencias de control:</h2>
  <table>
    <?php for ($f = 1; $f <= 3; $f++) : ?>
      <tr>
        <?php for ($c = 1; $c <= 5; $c++) : ?>
          <td><?php echo $f,$c ?></td>
        <?php endfor ?>
      </tr>
    <?php endfor ?>
  </table>
</html>
```

Validación de Datos en la misma página

- **Ej6:** Crea un formulario en un único fichero php, con estructura Html, que solicite el **nombre** y los **apellidos**. En caso de que no se cubra alguno de los campos, muestra un **mensaje de error** debajo del control correspondiente. Muestra **otro msj** con los **datos correctamente cubiertos**.
- **Requisitos:**
 - ✓ Verifica el método de envío
 - ✓ Comprueba si se ha cubierto el nombre y los apellidos.
 - ✓ Corrige el valor de los campos, eliminando espacios en blanco redundantes y cualquier carácter de código.
 - ✓ **Muestra el valor del campo correctamente cubierto.**
 - Añade como **value** de cada campo: `value='<?php echo $campo;?>`

Introduzca su nombre

Introduzca su apellido

Introduzca su nombre, por favor

Introduzca su apellido, por favor

Introduzca su nombre

Introduzca su apellido

Introduzca su apellido, por favor

Introduzca su nombre

Introduzca su apellido

Gracias por introducir sus datos, Pepe!!