

A2_T4. Formularis

Los formularios web permiten que el usuario envíe información, a través de su navegador, al servidor para que sea procesada. Son el equivalente a un formulario de papel en el que el usuario rellena varios campos con su información, al pulsar el botón de envío dicha información se transmite al servidor. Los formularios web pueden contener diferentes campos de entrada (texto, checkbox, fichero, ...) y un botón que activa el envío de datos al servidor. Opcionalmente también pueden tener un botón que limpia los campos del formulario.

Una vez transmitidos los datos al servidor, se ejecuta la página web dinámica que está asociada con el formulario, y allí se procesará la información recibida.

Aunque los formularios web tradicionales tienen algunos campos de entrada básica que también están presentes en la mayoría de interfaces gráficas, no presentan una gran variedad de elementos. La norma XForms proporciona una nueva tecnología para implementar formularios web.

Sintaxis de un formulario web

En HTML existe el elemento **form** que nos permitirá definir un formulario con sus campos de entrada, un botón de envío y opcionalmente uno de borrado y la URL a la que el navegador enviará los datos.

Un ejemplo:

```
<html>
<body>
  <form action="ejemplo.php" method="get">
    Color: <input name="color" type="text" />
    Sabor: <input name="sabor" type="text" />
    <input type="submit" />
  </form>
</body>
</html>
```

Conviene apreciar:

- Que el elemento **form** utiliza el atributo **action** para especificar a quién se enviarán los datos y el atributo **method** para indicar cómo.
- Que, además de una etiqueta, se han especificado dos campos en el formulario. Uno es un campo de entrada de texto identificado por el nombre 'user' y el otro es el propio botón de envío.

La [especificación del W3C sobre formularios](#) enumera los [distintos campos de entrada](#).

El [envío de los datos de un formulario al servidor](#) puede realizarse mediante el método [GET](#) o el método [POST](#). Con el primero los datos pasados quedan expuestos en la URL, visible para el usuario, y sólo debería utilizarse para formularios que no tengan efectos secundarios (por ejemplo una búsqueda no suele alterar la base de datos). El método POST debe utilizarse cuando los datos transmitidos provocan cambios en el servidor (un nuevo registro, una actualización o borrado) o no se quiere que se vean los parámetros pasados. Cuando se cargan ficheros debe utilizarse el método POST.

Procesado de formularios en PHP

Una página PHP puede recibir datos desde un formulario a través de las variables `$_GET`, `$_POST` o `$_REQUEST`. La primera contendrá aquellos campos que se han pasado mediante una petición GET, la segunda los que se han pasado mediante una petición POST y la tercera contiene los campos independientemente del método utilizado.

Así, para recuperar los valores recibidos por el ejemplo anterior, el fichero **ejemplo.php** podría contener:

```
<?php
    echo '<p>Color: ' . $_GET['color'] . '</p>';
    echo '<p>Sabor: ' . $_GET['sabor'] . '</p>';
?>
```

Un caso que merece especial interés es la [carga de ficheros](#) mediante un campo de entrada tipo **file**. En este caso es necesario definir en el formulario que se utilizará el método POST y la codificación multipart/form-data que permite enviar datos binarios.

```
<!-- El tipo de codificación de datos, enctype, se DEBE especificar como a continuación -->
<form enctype="multipart/form-data" action="__URL__" method="POST">
  <!-- MAX_FILE_SIZE debe preceder el campo de entrada de archivo -->
  <input type="hidden" name="MAX_FILE_SIZE" value="30000" />
  <!-- El nombre del elemento de entrada determina el nombre en el array $_FILES -->
  Enviar este archivo: <input name="userfile" type="file" />
  <input type="submit" value="Send File" />
</form>
```

En el lado del servidor el array **\$_FILES[]** contendrá toda la información sobre los ficheros que se han subido. Cuando se sube un fichero al servidor este se crea en el directorio temporal definido en la configuración de PHP, y antes de que termine la ejecución del script debe moverse/copiarse a su ubicación definitiva. Al terminar la ejecución del script se borrará el fichero temporal.

Ampliar información:

- [Wikipedia: Formulario web](#)
- [Traducción W3C HTML4: Formularios](#)
- [PHP: Uso de Formularios](#)
- [Wikipedia: XForms](#)
- [PHP: Manejo de XForms](#)