

# UT4. Formularios Web recuperación de información proveniente del cliente

DAW - Desarrollo Web Entorno Servidor  
Fernando Galindo

1. Formularios
2. Manejo de ficheros
3. Redirecciones y códigos de estado

# 1. Formularios

HTML surgió para mostrar información al cliente y no para que el usuario envíe datos al servidor. Este requisito ganó fuerza, con el tiempo, surgiendo los formularios.

Los formularios es un tipo de código HTML que nos permite **recabar información** y enviarla a un servidor Web para su posterior tratamiento.

# 1. Formularios

Los formularios HTML van encerrados siempre entre las etiquetas `<FORM>` `</FORM>`. Dentro de un formulario se incluyen los elementos sobre los que puede actuar el usuario, principalmente usando las etiquetas `<INPUT>`, `<SELECT>`, `<TEXTAREA>` y `<BUTTON>`.

El atributo **action** del elemento FORM indica la página a la que se le enviarán los datos del formulario.

En nuestro caso se tratará de un guión PHP.

# 1. Formularios

El atributo `method` especifica el método para enviar información al formulario

- **get**: con este método los datos del formulario se agregan al URI utilizando un signo de interrogación "?" como separador.
- **post**: con este método los datos se incluyen en el cuerpo del formulario y se envían utilizando el protocolo HTTP.

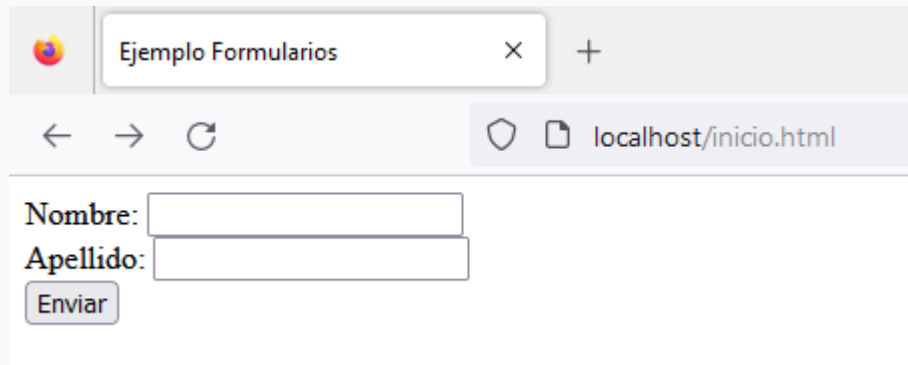
**Cuidado con el método get y enviar información sensible**

En conexiones **NO CIFRADAS**, ambos métodos son igual de inseguros

# 1. Formularios

## Ejemplo sencillo, petición datos

```
<html>
<head>
  <title>Ejemplo Formularios</title>
</head>
<body>
  <form action="/servidor.php" method="post">
    <label>Nombre:</label>
    <input type="text" name="nombre"><br><br>
    <label>Apellido:</label>
    <input type="text" name="apellido"><br><br>
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```

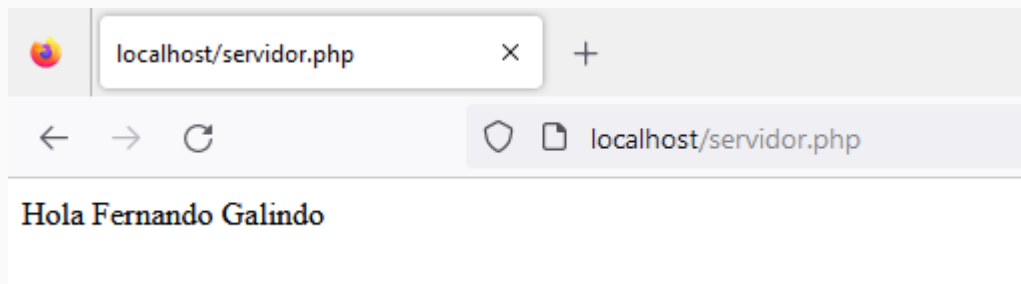


A screenshot of a web browser window. The title bar shows 'Ejemplo Formularios' with a close button. The address bar shows 'localhost/inicio.html'. The page content displays a form with two text input fields. The first field is labeled 'Nombre:' and the second is labeled 'Apellido:'. Below the second field is a submit button labeled 'Enviar'.

# 1. Formularios

## Ejemplo sencillo, tratamiento de los datos

```
<?php
$nombre = $_POST['nombre'];
$apellido = $_POST['apellido'];
echo "Hola ".$nombre." ".$apellido;
?>
```



## Probemos el funcionamiento cambiando los datos por GET



# 1. Formularios

Atributos type de la etiqueta input: [link1](#) & [link2](#)

<input type="button">

<input type="checkbox">

<input type="color">

<input type="date">

<input type="datetime-local">

<input type="email">

<input type="password">

<input type="radio">

<input type="range">

<input type="reset">

<input type="search">

<input type="submit">

<input type="tel">

<input type="text">

<input type="time">

<input type="url">

<input type="file">

<input type="hidden">

<input type="image">

<input type="month">

<input type="week">

<input type="number">



# 1. Formularios

## La etiqueta label nos agrupa los elementos

```
<html>
<head>
  <title>Ejemplo Label</title>
</head>
<body>
<h1> Con Label</h1>
<form>
  <input type="checkbox" id="modulo_lmsg" name="modulo_lmsg" value="lmsg">
  <label for="modulo_lmsg"> Lenguajes de Marcas</label><br>
  <input type="checkbox" id="modulo_edes" name="modulo_edes" value="edes">
  <label for="modulo_edes"> Entornos de desarrollo</label><br>
  <input type="checkbox" id="modulo_dews" name="modulo_dews" value="dews">
  <label for="modulo_dews"> Desarrollo Web Entorno Servidor</label>
</form>
</body>
</html>
```

# 1. Formularios




## La etiqueta label nos agrupa los elementos

```
<html>
<head>
  <title>Ejemplo Label</title>
</head>
<body>
<h1> Sin Label</h1>
<form>
  <input type="checkbox" id="modulo_lmsg" name="modulo_lmsg" value="lmsg">
Lenguajes de Marcas<br>
  <input type="checkbox" id="modulo_edes" name="modulo_edes" value="edes">
Entornos de desarrollo<br>
  <input type="checkbox" id="modulo_dews" name="modulo_dews" value="dews">
Desarrollo Web Entorno Servidor
</form>
</body>
</html>
```

## 2. Redirecciones y códigos de estado

Las llamadas a páginas Web tienen un código de estado que indica si una petición se ha realizado de forma satisfactoria

- Respuestas informativas (100–199)
- Respuestas satisfactorias (200–299)
- Redirecciones (300–399)
- Errores de los clientes (400–499)
- Errores de los servidores (500–599)

Estado	Método	Dominio
302	GET	 localhost
200	GET	 localhost
404	GET	 localhost

<https://developer.mozilla.org/es/docs/Web/HTTP/Status>

## 2. Redirecciones y códigos de estado

La función `header` nos permite realizar redirecciones (3xx) que quedan registradas, de igual forma que los accesos no autorizados a determinadas páginas queden registrados con los códigos de error

Podemos incluir información de la respuesta en la función `header`

```
header('HTTP/1.0 401 Unauthorized');
```

O bien utilizar la siguiente función de forma más simple

```
http_response_code('401');
```

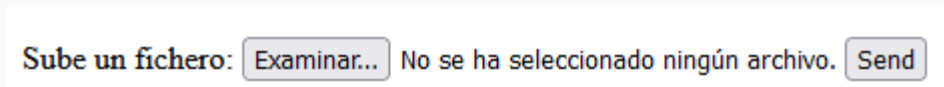
## 2. Redirecciones y códigos de estado

```
<?php
if (!isset($_SERVER['PHP_AUTH_USER'])) {
    header('WWW-Authenticate: Basic Realm="Contenido restringido");
    header('HTTP/1.0 401 Unauthorized');
    echo "Usuario No Valido";
    exit;
}
?>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<title>Ejercicio: Función header para autenticación HTTP</title>
<link href="dwes.css" rel="stylesheet" type="text/css">
</head>
<body>
<?php
echo "Nombre de usuario: ".$_SERVER['PHP_AUTH_USER']."<br />";
echo "Contraseña: ".$_SERVER['PHP_AUTH_PW']."<br />";
?>
</body>
</html>
```

### 3. Manejo de ficheros

Los tipos de datos que podemos tener a la hora de enviar ficheros a un servidor son los ficheros

```
<input type="file" name="fichero" />
```



Resulta interesante poder indicar los tipos de ficheros que vamos a aceptar

```
<input type="file" name="fichero"  
accept="application/pdf,.pdf" />
```

Podemos indicar extensiones o bien tipos MIME

<https://www.iana.org/assignments/media-types/media-types.xhtml>

### 3. Manejo de ficheros

Los tipos aceptados en el formulario no son de obligado cumplimiento por parte del usuario y puede saltarlos de forma fácil

A mayores indicamos cómo se van a codificar los archivos para enviarlos al servidor. [Revisar documentación](#)

```
<form action="" method="post" enctype="multipart/form-data">
<label>Sube un fichero:
<!-- MAX_FILE_SIZE indica el tamaño máximo -->
<input type="hidden" name="MAX_FILE_SIZE" value="30000" />
<input type="file" name="fichero" accept="
application/pdf,.pdf" /></label>
<input type="submit" value="Subir" />
</p>
</form>
```

### 3. Manejo de ficheros

Una vez construido el formulario podemos tratar el fichero desde nuestra aplicación PHP

El fichero lo recibimos en la variable `$_FILES["fichero"]` se trata de una variable global y a su vez un array asociativo

```
array (size=5)
  'name' => string 'leer.txt' (length=8)
  'type' => string 'text/plain' (length=10)
  'tmp_name' => string 'C:\xampp\tmp\php2DBA.tmp' (length=24)
  'error' => int 0
  'size' => int 1910
```



### 3. Manejo de ficheros

`MAX_FILE_SIZE` dentro del formulario como campo oculto nos limita el tamaño que podemos subir al servidor, que no debemos confundir con la directiva `MAX_FILE_SIZE` del fichero `php.ini`

Esto varía el campo `$_FILES["fichero"]["error"]`

```
array (size=5)
  'name' => string 'FicheroGrande.pdf' (length=28)
  'type' => string '' (length=0)
  'tmp_name' => string '' (length=0)
  'error' => int 2
  'size' => int 0
```

<https://www.php.net/manual/en/features.file-upload.errors.php>

### 3. Manejo de ficheros

Realiza un formulario y una aplicación Web en la que se muestran diferentes tipos de error en función del “error” a la hora de manejar el fichero

Fichero subido correctamente, fichero excede de la directiva PHP ini y fichero excede de la directiva MAX\_FILE\_SIZE (se mide en bytes)

Sería conveniente realizar más comprobaciones: tamaño máximo, tipo del fichero, etc. Antes de guardar el fichero en el sistema de archivos del servidor.

<https://www.php.net/manual/en/features.file-upload.php>

### 3. Manejo de ficheros

Una vez realizadas las validaciones el siguiente código, almacena el fichero subido en el sistema de archivos del servidor

```
$error=$_FILES["fichero"]["error"];  
if ($error == UPLOAD_ERR_OK) {  
    $tmp_name = $_FILES["fichero"]["tmp_name"];  
    // basename() may prevent filesystem traversal  
attacks;  
    $name = basename($_FILES["fichero"]["name"]);  
    move_uploaded_file($tmp_name, $name);  
}
```

### 3. Manejo de ficheros

Los ficheros se almacenan con el nombre con el que el usuario tenía identificado el archivo

Problema no podemos almacenar dos ficheros con el mismo nombre de diferentes usuarios

Tenemos que pensar en alternativas

<https://www.php.net/manual/en/function.uniqid.php>