



### TEMAS A DESARROLLAR

Algo de HTML.

Formularios en HTML.

Métodos de envío.

Administración de datos desde PHP.



**Algo de HTML**

### Algo de HTML

HTML, Lenguaje de Marcado de Hipertextos (HyperText Markup Language), es la pieza más básica para la construcción de la web y se usa para definir el sentido y estructura del contenido en una página web.

Además de HTML se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript).

**"Hipertexto"** se refiere a los enlaces que conectan las páginas web entre sí, ya sea dentro de un mismo sitio web o entre diferentes sitios web.

HTML usa "marcado" para anotar textos, imágenes y otro contenido para ser mostrado en un navegador web. El marcado en HTML incluye "elementos" especiales tales como <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, <img>, <aside>, <audio>, <nav>, <output>, <video>, <ul>, <ol>, <li>, y muchos otros más.

### Algo de HTML

Un elemento HTML se separa de otro texto en un documento por medio de "**etiquetas**", las cuales consisten en elementos rodeados por "<" y ">".

Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Programación Avanzada</title>
  </head>
  <body>
    <h1>Práctica de Programación Avanzada</h1>
    <p>Éste es un ejemplo sencillo de un párrafo.</p>
  </body>
</html>
```

## Estructura de un documento HTML

La estructura de un documento HTML consta de dos secciones: el **head** y el **body**. Es **head** es la parte del documento que primero llega al navegador web y contiene instrucciones que le indican cómo debe mostrar el contenido (que llegará luego en el **body**). Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Programación Avanzada</title>
    <meta charset="UTF-8">
    <meta name="title" content="Título de la WEB">
    <meta name="description" content="Site Programación Avanzada">
    <link href="/styles.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
```

## Estructura de un documento HTML

El contenido (la parte visible del documento) se describe dentro de las etiquetas: **<body> </body>**. Su contenido es mucho mas extenso ya que se incluye todo lo que debe mostrarse en la página web.: Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Programación Avanzada</title>
  </head>

  <body>
    <h1>Práctica de Programación Avanzada</h1>
    <p>Éste es un ejemplo sencillo de un párrafo.</p>
  </body>
</html>
```



## Formulario en HTML

### Formularios en HTML

Los formularios sirven para **recolectar información proporcionada por los visitantes del sitio** la cual luego **es enviada al servidor**. Para su funcionamiento el formulario HTML debe ser acompañado de un código del lado servidor que se encargará de recibir y procesar la información.

Un formulario **<form>** es básicamente un **contenedor de controles**. Cada control está pensado para recolectar información ingresada por los usuarios, en formas que pueden ir desde líneas de texto a subida de archivos, pasando por opciones, fechas, contraseñas y otras más.

Una vez que el usuario han rellenado el formulario con los datos puede enviarlo de regreso al servidor para que un script administre la información recolectada.





## Métodos de envío

### Formularios en HTML

El atributo **method** especifica **cómo enviar los datos de formulario**. Existen dos métodos para enviar los datos desde el navegador hacia el servidor: **get** y **post**.

Cuando se utiliza el método **get**, los datos del formulario se envían **como variables de URL** anexándose a la misma de la forma **nombre=valor**.

Algunas características:

- Agrega datos de formulario a la URL en pares de nombre / valor.

- La longitud de una URL es limitada (aproximadamente 3000 caracteres).

- No se debe utilizar GET para enviar datos confidenciales! (será visible en la URL).

- Útil para envíos de formularios donde un usuario desea marcar el resultado.

### Formularios en HTML

Ejemplo:

```
<form action="procesoFormulario.php" method="get" >
```

```
Nombre: <input type="text" name ="nombre" >
```

```
Apellido: <input type="text" name ="apellido" >
```

```
<input type="submit" value="Enviar el formulario">
```

```
</form>
```

Al enviar el formulario la URL quedará de la siguiente forma:

```
http://localhost/procesoFormulario.php?nombre=Federico&apellido=Bonnet
```

### Formularios en HTML

Cuando definimos un formulario para que envíe sus datos por el método **post** (**method=post**), éstos **no se agregan a la URL sino que se anexan en el cuerpo de la solicitud HTTP**.

De esta forma no tenemos límite en el tamaño y tipo de datos a enviar y no se exponen los datos en la URL (por lo cual una contraseña no quedaría expuesta).

**¡¡¡Lo cual no significa que los datos se cifren, solamente no son expuesto en la barra de navegación!!!**

## Formularios en HTML

Ejemplo:

```
<form action="procesoFormulario.php" method="post">  
  Nombre: <input type="text" name ="nombre" >  
  Apellido: <input type="text" name ="apellido" >  
  <input type="submit" value="Enviar el formulario">  
</form>
```

Los datos se envían de la siguiente forma:

```
POST / HTTP/1.1  
Host: localhost  
Content-Type: application / x-www-form-urlencoded  
Content-Length: 30  
  nombre=Federico & apellido=Bonnet
```

### Formularios en HTML

El atributo **action** determina **el nombre del script que procesará los datos** recabados por el formulario(a que script se envían los datos). En nuestro ejemplo:

```
<form action="procesoFormulario.php" method="get">  
  Nombre: <input type="text" name ="nombre" >  
  Apellido: <input type="text" name ="apellido" >  
  <input type="submit" value="Enviar el formulario">  
</form>
```

El script **procesoFormulario.php** será el encargado de recibir los datos enviados desde el navegador a través del formulario. **Deberemos programar en éste todas las instrucciones para administrar esa información.**



**Administración de datos desde  
PHP.**

### Administrar los datos desde PHP

Para procesar los datos provenientes de un formulario con PHP lo primero que deberemos hacer es recuperar lo mismo.

Para ésto PHP define una serie de arreglos que, **dependiendo del método** que se utilizó en el formulario para enviar los datos, nos permitirán recuperar los mismos.

Estos arreglos son: **\$\_GET**, **\$\_POST** y **\$\_REQUEST**.

Si en el formulario definió: **method=get**, los datos los recuperará desde: **\$\_GET**.

Si en el formulario definió: **method=post**, los datos los recuperará desde: **\$\_POST**.



### Administrar los datos desde PHP

Ya sea que que utilice `method=get` o `method=post`, los datos siempre estarán disponibles en el arreglo: `$_REQUEST`.

Ejemplo:

```
<?php
    print_r ( $_GET );
    echo ( $_GET[ 'nombre' ] );
    echo ( $_GET[ 'apellido' ] );
?>
```

```
<?php
    print_r ( $_POST );
    echo ( $_POST[ 'nombre' ] );
    echo ( $_POST[ 'apellido' ] );
?>
```

### Administrar los datos desde PHP

Ya sea que los datos del formulario se envíen por **get** o por **post**, php definirá un arreglo de nombre `$_REQUEST` que contendrá la misma definición de datos sin importar el método de envío.

Arreglo: **`$_REQUEST`**.

Si en el formulario definió: **`method=get`**, los datos los recuperarán también con: **`$_REQUEST`**.

Si en el formulario definió: **`method=post`**, los datos los recuperarán también con: **`$_REQUEST`**.