

TEMA 8. GENERACIÓN DINÁMICA DE PÁGINAS WEB INTERACTIVAS.

Objetivos

- Reconocer las ventajas de unir tecnologías del lado del cliente y del servidor.
- Identificar librerías y tecnologías para generar páginas web que incluyan interacción con el usuario en forma de advertencias y peticiones de confirmación.
- Utilizar estas tecnologías para generar páginas web que verifiquen formularios.
- Utilizar estas tecnologías para generar páginas que incluyan modificación dinámica de su contenido y estructura.

8.1.- Introducción a la Generación Dinámica de Páginas Web Interactivas.

Página Web interactiva es una página que permite variar el aspecto y comportamiento en función de decisiones que toma el usuario a través de la interfaz gráfica de la página.

Se puede utilizar:

- Código de script que se ejecuta en el navegador.
- Enviar una petición al servidor. Se ejecuta el código en el servidor y genera una nueva página Web que envía al navegador. Al tratarse de una página Web dinámica, en función de los datos introducidos por el usuario, la página se verá modificada mostrando información definida en la programación.

Tipos de Código que se ejecutan en el cliente:

- **HTML.** Base de la página web que describe la estructura y el contenido que es interpretada por el navegador.
- **Javascript.** permite configurar la interactividad y el dinamismo de la página web. El código javascript se ejecuta cuando ocurre un evento que tenga asociada una función.
- **CSS.** Encargado de definir la presentación de los datos que se indican en el código HTML.
- **XML.** Lenguaje de marcas extensible que permite el intercambio de datos.
- **Otros:** Applets de Java, VBScript, etc..

Tipos de Código que se ejecutan en el servidor:

- **CGI.** Programa que se ejecuta en el servidor y envía datos al navegador.
- **PERL.** El código se ejecuta directamente en el servidor y permite realizar llamadas a otros subprogramas.
- **ASP.** Tecnología de Microsoft que permite generar páginas web dinámicas en el servidor.
- **PHP.** Lenguaje interpretado que se ejecuta en el lado del servidor y se envía al navegador en forma de HTML con las siguientes características: Libre y abierto, librerías de funciones, se embebe en el código HTML.
- **JSP.** Permite crear páginas con programación Java. Es un lenguaje multiplataforma que se ejecuta en servidores con la Máquina Virtual instalada.

8.2.- Librerías y Tecnologías relacionadas.

Una librería o biblioteca es un conjunto de subprogramas que sirven para programar una aplicación y que pueden ser llamados desde otro programa principal.

Una interfaz de programación de aplicaciones (API), es un conjunto de funciones y procedimientos que proveen de bibliotecas para ser utilizado por otro programa de forma transparente al programador.

Un framework emplea módulos concretos que facilitan desarrollar , agrupar y organizar los componentes del nuevo software.

Tecnologías y Librerías relacionadas con PHP.

PHP es un lenguaje que funciona en diferentes plataformas (Windows, Linux, Unix,etc.) que es compatible con casi todos los servidores que se utilizan en la actualidad (Apache, IIS, etc.). Se puede descargar gratis desde los recursos oficiales de PHP. Es fácil de aprender y se ejecuta en el lado del servidor.

En PHP podemos hacer uso de librerías utilizando la palabra reservada **include**:

```
i ncl ude "fi chero. php";
```

Ejemplo de uso de una librería:

calculaHipotenusa.php

```
<?php
    i ncl ude "funci ones. php";
    $cateto1 = 10;
    $cateto2 = 5;
    $hi potenusa = cal cul aHi potenusa($cateto1, $cateto2);
    echo "La hi potenusa del tri ángulo es ".$hi potenusa;
?>
```

funciones.php

```
<?php
    functi on cal cul aHi potenusa($arg1, $arg2){
        $hi potenusa = sqrt($arg1*$arg1+$arg2*$arg2);
        return $hi potenusa;
    }
?>
```

8.3.- Generación dinámica de páginas Interactivas.

Podemos ejecutar scripts de navegador para solicitar otra página al servidor capturando eventos con la redirección. Con la petición de la nueva página se puede enviar variables a través de la URL.

Ejemplo:

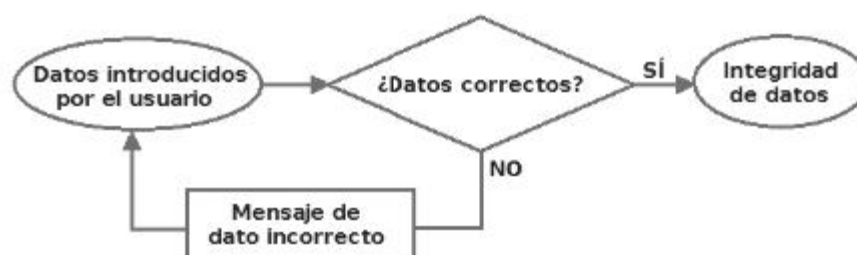
```
<html >
<head>
  <title>Páginas Interactivas PHP</title>
  <script language="text/javascript">
    function envio(){
      if(confirm('Pulsa una opción?')){
        location.href='destino.php?id=1';
      }else{
        location.href='destino.php?id=2';
      }
    }
  </script>
</head>
<body>
  <p onclick="envio()">Destino Página nueva</p>
</body>
</html >
```

destino.php

```
<html >
  <head>
    <title>Destino PHP</title></head>
  <body>
    <?php
      $valor=$_GET["id"];
      echo"Estas en Destino. Valor: ".$valor;
    ?>
  </body>
</html >
```

8.4.- Obtención Remota de Información.

Para recuperar campos en el servidor, se utilizan los formularios de HTML. Se envía el formulario desde el cliente al servidor, pero al recuperar los valores no se puede garantizar que el tipo de dato coincida con el que se espera obtener por lo que se deben de comprobar previamente los valores que llegan desde el navegador.



Para validar y filtrar datos en el servidor utilizamos las siguientes sentencias:

Nombre del Filtro	Descripción del Filtro
<code>filter_var()</code>	Filtra una variable con un filtro específico.
<code>filter_has_var()</code>	Comprueba si existe una variable de un tipo concreto.
<code>filter_input()</code>	Obtiene un campo de entrada y un filtro para este.

Ejemplo: `filter_has_var(tipoEntrada, variable)`

Los diferentes tipos de entrada son:

- `INPUT_GET`
- `INPUT_POST`
- `INPUT_COOKIE`
- `INPUT_SERVER`

```
<?php
if (filter_has_var(INPUT_GET, "email")) {
    echo("Email encontrado");
} else {
    echo("Email no encontrado");
}
?>
```

Ejemplo: `filter_var(variable, tipoFiltro)`

Los diferentes tipos de validación son:

- `FILTER_VALIDATE_BOOLEAN` Valida boolean
- `FILTER_VALIDATE_EMAIL` Validat e-mail
- `FILTER_VALIDATE_FLOAT` Valida float
- `FILTER_VALIDATE_INT` Valida entero
- `FILTER_VALIDATE_IP` Valida dirección IP
- `FILTER_VALIDATE_REGEXP` Valida expresión regular
- `FILTER_VALIDATE_URL` Valida URL

```
<?php
$email = "mario@ono.com";

if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
    echo("$email es una dirección de correo válida");
} else {
    echo("$email no es una dirección de correo válida");
}

?>
```

Ejemplo: filter_input(tipoEntrada, variable, tipoValidación)

Los diferentes tipos de entrada son:

- INPUT_GET
- INPUT_POST
- INPUT_COOKIE
- INPUT_SERVER

Los diferentes tipos de validación son:

- FILTER_VALIDATE_BOOLEAN Valida boolean
- FILTER_VALIDATE_EMAIL Valida e-mail
- FILTER_VALIDATE_FLOAT Valida float
- FILTER_VALIDATE_INT Valida entero
- FILTER_VALIDATE_IP Valida dirección IP
- FILTER_VALIDATE_REGEXP Valida expresión regular
- FILTER_VALIDATE_URL Valida URL

```
<?php
if (filter_input(INPUT_GET, "email", FILTER_VALIDATE_EMAIL)) {
    echo("El email es valido");
} else {
    echo("El email no es valido");
}
?>
```

8.5.- Modificación de la estructura de la Página Web.

Mediante el acceso a la estructura de árbol DOM es posible modificar la estructura de una página Web sin que el código tenga que salir del propio navegador.

Ventaja: La flexibilidad que supone a la hora de generar páginas interactivas tanto a nivel visual como funcional.