Unidad 6 - Programación del lado del servidor


* @var boolean
define('PSI_INTERNAL_XML', false);
if (version_compare("5.2", PHP_VERSION, ">")) {
die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
die("phpSysInfo requires the pcre extension
properly.");
}
require_once APP_ROOT.'/includes/autol
// Load configuration
require_once APP_ROOT.'/config.php';
if (!defined('PSI_CONFIG_FILE') || !define
$tpl = new Template("/templates/html/erro
echo $tpl->fetch();
die();
javascript
strtolower(
php

PHP (Hypertext Preprocessor)

Es un lenguaje de scripting de código abierto ampliamente usado para el desarrollo de aplicaciones web del lado del servidor. Su principal ventaja es que permite generar contenido dinámico en páginas web.

Cuando un navegador solicita una página con código PHP, el servidor interpreta el código PHP y genera una respuesta en HTML, que luego es enviada al navegador.

Sintaxis básica

PHP es un lenguaje de scripting que se inserta en el código HTML usando etiquetas especiales **<?php ... ?>.**

Un archivo PHP suele tener la extensión **.php** y puede contener tanto código HTML como PHP.

Todas las sentencias deben de terminar con **;**

echo sirve para imprimir algo en la página, ya sea una cadena o el valor de una variable.

```php
<?php
echo "Hola, mundo!";
?>
```

Comentarios

```php
<?php
    echo 'Esto es una prueba'; // Comentario de una sola línea
    /* Esto es un comentario multilínea
       y otra línea de comentarios */
    echo 'Esto es otra prueba';
    echo 'Una prueba final'; # Comentario de una sola línea
?>
```

Variables

Las variables en PHP se definen con el símbolo **$** seguido del nombre de la variable. No es necesario declarar el tipo de la variable, ya que PHP es un lenguaje de tipado débil.

Para saber el tipo de dato de una variable puede usarse la función **gettype(**variable**).**

Para comprobar el tipo de dato se puede usar la función is_tipo. Por ejemplo **is_int(**variable**)** o **is_string(**variable**).**

```php
<?php

$nombre = "Ana";
$edad = 30;

?>
```

Constantes

Las constantes son valores que no cambian a lo largo de la ejecución del script. Se definen con la función **define(nombre, valor).**

```php
<?php

define("PI", 3.1416);
echo PI;

?>
```

Operadores

Aritméticos: **+, -, *, /, %** para operaciones matemáticas básicas.

Asignación: **=, +=, -=**, etc., para asignar valores a variables.

Comparación: **==, !=, ===, !==, >, <, >=, <=** para comparar valores.

Lógicos: **&&, ||, !** para realizar operaciones lógicas.

Estructuras de control

Las estructuras de control en PHP son fundamentales para controlar el flujo del programa.

- if
- switch
- while
- for
- do-while
- foreach

Arreglos

PHP soporta arreglos (arrays) para almacenar múltiples valores en una sola variable. Hay arreglos indexados, asociativos y multidimensionales.

```php
<?php
//Arreglo indexado
$frutas = array("manzana", "naranja", "plátano");
echo $frutas[0];  // Imprime "manzana"

?>
```

Arreglos asociativos

Los arreglos asociativos en PHP son una estructura de datos que permite almacenar pares de clave-valor, donde cada valor se asocia con una clave única que es una cadena de texto en lugar de un índice numérico. Este tipo de arreglo es útil cuando necesitas acceder a los valores usando nombres en lugar de índices.

```php
<?php
// Ejemplo usando array()
$persona = array(
    "nombre" => "Ana",
    "edad" => 30,
    "profesion" => "Ingeniera"
);

// Ejemplo usando corchetes []
$persona = [
    "nombre" => "Ana",
    "edad" => 30,
    "profesion" => "Ingeniera"
];

?>
```

Recorrer un Arreglo Asociativo

Para recorrer un arreglo asociativo y acceder a cada clave y valor, se utiliza el ciclo foreach.

```php
<?php
foreach ($persona as $clave => $valor) {
    echo "$clave: $valor <br>";
}
?>
```

Funciones

PHP permite definir funciones para reutilizar código. Las funciones se definen con **`function nombre_funcion() { ... }`** y se llaman usando el nombre de la función.

```php
<?php
function saludar($nombre) {
    return "Hola, $nombre";
}
echo saludar("Ana");  // Imprime "Hola, Ana"
?>
```

Inclusión de Archivos

PHP permite incluir archivos externos usando **include** o **require**. Esto es útil para separar código común, como funciones, encabezados o pies de página, en archivos separados.

include emitirá una **advertencia** si no puede encontrar un archivo, éste es un comportamiento diferente al de **require**, el cual emitirá un **error fatal**.

```php
<?php
include 'header.php';
echo "Contenido principal de la página.";
include 'footer.php';
?>
```