

Análisis de Sistemas

Materia:
Programación Web II

Docente contenidista: ROLDÁN, Hernán

Revisión: Coordinación



Contenido

Introducción a PHP: breve historia	3
¿Qué puede hacerse con PHP?.....	5
Sitios estáticos vs sitios dinámicos.....	6
Lenguajes de programación del lado del servidor.....	7
Instalación de PHP (instalación independiente).....	10
Instalación de entornos locales que soportan PHP.....	11
Editores online que soportan PHP	12
Sintaxis Básica PHP	13
Comentarios	17
Variables	18
Bibliografía	22
Para ampliar la información	22

Clase 1



¡Te damos la bienvenida a la materia
Programación Web II!

En esta clase vamos a ver los siguientes temas:

- Introducción a PHP: breve historia.
- ¿Qué puede hacerse con PHP?
- Sitios estáticos vs sitios dinámicos.
- Lenguajes de programación del lado del servidor.
- Instalación de PHP (instalación independiente).
- Instalación de entornos locales que soportan PHP.
- Editores online que soportan PHP.
- Sintaxis Básica PHP.
- Comentarios.
- Variables.

Programación PHP

Introducción a PHP: breve historia

PHP fue creado en 1994 por Rasmus Lerdorf como un conjunto de herramientas para el seguimiento de visitantes de su sitio web personal, de ahí las siglas iniciales de "Personal Home Page Tools". En 1995, Andi Gutmans y Zeev Suraski reescribieron el núcleo de PHP, lo que lo convirtió en un lenguaje más eficiente y robusto.

Desde entonces, PHP se ha utilizado ampliamente en la creación de sitios web dinámicos y ha sido adoptado por muchas empresas y organizaciones de todo el mundo.

A pesar de su evolución, el nombre "PHP" se ha mantenido como un recordatorio de sus orígenes, pero haciendo referencia al acrónimo recursivo de "PHP Hypertext Preprocessor".

PHP es conocido por su facilidad de uso y su capacidad para integrarse con otros lenguajes y sistemas, como HTML, CSS, JavaScript y bases de datos como MySQL. Además, existe una gran comunidad de usuarios y desarrolladores de PHP, lo que significa que hay muchos recursos disponibles para aprender y resolver problemas.

Algunos de los usos más comunes de PHP incluyen:

- Desarrollo de sitios web dinámicos.
- Creación de aplicaciones web.
- Integración con bases de datos.
- Generación de contenido dinámico en tiempo real.
- Procesamiento de formularios y envío de correos electrónicos.

Algunas de las empresas conocidas que utilizan PHP son:

- Facebook
- Wikipedia
- Yahoo
- Etsy
- Tumblr
- WordPress
- Netflix
- Lyft
- Slack
- Square



Imágenes extraídas de la web

Estas empresas han adoptado PHP debido a su facilidad de uso, flexibilidad y capacidad para integrarse con otros sistemas y lenguajes.

Además, dada la gran comunidad de usuarios y desarrolladores de PHP implica que hay muchos recursos disponibles para aprender y resolver problemas.

En resumen, PHP es un lenguaje de programación versátil y popular que ha demostrado ser una herramienta valiosa para la creación de sitios web dinámicos y aplicaciones web.

¿Qué puede hacerse con PHP?

Con PHP, se pueden hacer muchas cosas, incluyendo:

- Desarrollo de sitios web dinámicos.
- Procesamiento de formularios.
- Conexión a bases de datos.
- Creación de scripts de administración de contenido.
- Generación de páginas dinámicas con contenido almacenado en una base de datos.
- Creación de aplicaciones web como tiendas en línea, foros, blogs, etc.
- Procesamiento de imágenes y archivos multimedia.

Estas son solo algunas de las muchas cosas que se pueden hacer con PHP. Es uno de los lenguajes de programación más populares y ampliamente utilizados en el mundo para el desarrollo de aplicaciones web.

Sitios estáticos vs sitios dinámicos

Un sitio web estático es aquel cuyo contenido se muestra de la misma manera para todos los usuarios y no cambia con el tiempo.

Es un conjunto de archivos HTML, CSS y JavaScript que se sirven directamente al cliente desde un servidor web. Los sitios web estáticos son simples de crear y mantener, ya que no requieren una interacción compleja con el usuario ni una base de datos.

Sin embargo, su contenido no se puede personalizar para cada usuario y no admiten actualizaciones en tiempo real.

Por otro lado, un sitio web dinámico es aquel cuyo contenido puede cambiar en función de las interacciones del usuario y otros datos variables. Utiliza tecnologías de programación en el lado del servidor, como PHP, Python o Ruby, para generar contenido dinámicamente antes de enviarlo al cliente.

Los sitios web dinámicos pueden acceder a bases de datos para recuperar y mostrar información actualizada, permitiendo a los usuarios interactuar con formularios, iniciar sesión, realizar compras en línea y más.

También pueden adaptarse a las preferencias del usuario y personalizar el contenido según sus necesidades.

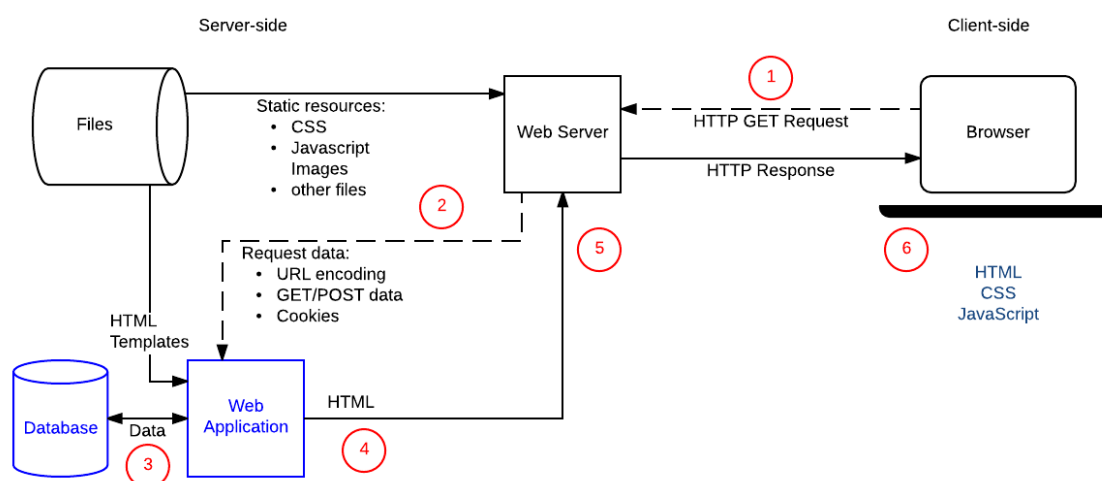


Imagen extraída de la web

Lenguajes de programación del lado del servidor

Un lenguaje de programación server-side (del inglés, "lado del servidor") es aquel que se utiliza para crear aplicaciones web que se ejecutan en el servidor. Esto significa que el código escrito en este lenguaje se ejecuta en el servidor, en lugar de en el navegador web del usuario.

Cuando un usuario solicita una página web que utiliza un lenguaje server-side, el servidor ejecuta el código y envía al usuario una versión "renderizada" de la página web. Por ejemplo, si el lenguaje server-side utilizado es PHP, el servidor ejecutará el código PHP y enviará al usuario una página HTML generada por el servidor.

El proceso en etapas que realiza un lenguaje server-side básicamente es el siguiente:

1. El cliente realiza una solicitud (por ejemplo, haciendo click en un enlace o enviando un formulario) a una página web alojada en un servidor.
2. El servidor recibe la solicitud y ejecuta el código server-side correspondiente al archivo que se ha solicitado (por ejemplo, un archivo PHP).
3. El código server-side procesa la solicitud, accede a la base de datos, manipula los datos, realiza cálculos y, en general, genera una respuesta dinámica que se ajusta a las necesidades del cliente.
4. La respuesta generada por el código server-side se envía de vuelta al cliente en forma de una página web renderizada que contiene HTML, CSS y JavaScript, pero que ha sido generada dinámicamente por el servidor a partir del código server-side.
5. El cliente recibe la página web renderizada y la muestra en el navegador web.

El siguiente código es un ejemplo sencillo de cómo se podría programar en PHP lo anteriormente citado:

Este proceso se repite cada vez que el cliente realiza una nueva solicitud a una página web alojada en el servidor, permitiendo que la página web sea actualizada y modificada dinámicamente en función de las acciones del usuario y de los datos que se almacenan en el servidor.

Los lenguajes de programación server-side se utilizan para desarrollar aplicaciones web dinámicas, donde el contenido de la página se genera en tiempo real en función de la interacción del usuario o de la información almacenada en una base de datos.

Los lenguajes de programación server-side más comunes son PHP, Python, Ruby, Java, y JavaScript (utilizado a menudo en su forma de Node.js).

Un ejemplo de programación del lado del servidor con PHP sería la creación de un formulario de registro de usuarios que, al enviar la información al servidor, éste la procesa y almacena en una base de datos.

Ejemplo de un formulario HTML:

```
<html>
<head>
  <title>Formulario PHP</title>
</head>
<body>
  <h1>Formulario de Registro</h1>
  <form action="registro.php" method="post">
    <label for="nombre">Nombre:</label>
    <input type="text" name="nombre" id="nombre" required>
    <br>
    <label for="email">Email:</label>
    <input type="email" name="correo" id="correo" required>
    <br>
    <label for="clave">Contraseña:</label>
    <input type="password" name="clave" id="clave" required>
    <br>
    <input type="submit" value="Registrar">
  </form>
</body>
</html>
```

Creación del archivo "registro.php" que procesa la información enviada por el formulario:

```
<?php

// Recupera los datos enviados en el formulario
$nombre = $_POST['nombre'];
$email = $_POST['correo'];
$clave = $_POST['clave'];

// Establece la conexión con la base de datos
$conexion = mysqli_connect('localhost', 'usuario_bd', 'clave_bd',
'basedatos');

// Si se genera un error durante la conexión muestra un mensaje y
detiene la ejecución del programa
if(mysqli_connect_errno()){
    echo 'Se produjo un error durante la conexión a la base de
datos.' . mysqli_connect_error();
    exit();
}

// Si no, inserta los datos del usuario en la base de datos
else{

// Consulta SQL
$query = "INSERT INTO usuarios (nombre, email, clave) VALUES
('$nombre', '$correo', '$clave')";

// Ejecuta la consulta SQL contra la base de datos
mysqli_query($conexion, $query);

// Cierra la conexión a la base de datos
mysqli_close($conexion);

// Redirige al usuario a una página de confirmación
header('Location: confirmacion.php');

}

?>
```

Este es solo un ejemplo básico de programación del lado del servidor con PHP. En aplicaciones más complejas se utilizan múltiples archivos PHP para manejar diferentes funciones y procesos, y se pueden integrar otros lenguajes de programación y tecnologías como bases de datos, frameworks, APIs, etc.

Instalación de PHP (instalación independiente)

Para usar PHP, se necesita tener instalado lo siguiente:

1. Un servidor web:

PHP se ejecuta en un servidor web, como Apache o Nginx, y es el servidor web quien se encarga de procesar los scripts PHP y enviar la respuesta al navegador del usuario.

2. PHP:

La versión más reciente de PHP puede ser descargada desde la página oficial de descarga: <https://windows.php.net/download/>.

3. Un navegador web:

Para visualizar las páginas generadas por PHP, se necesita un navegador web, como Google Chrome, Mozilla Firefox o Microsoft Edge.

4. Un editor de texto o entorno de desarrollo integrado (IDE):

Para escribir y editar los scripts PHP. Algunos ejemplos de editores de texto son Notepad++ en Windows o Sublime Text en Windows, macOS y Linux. Algunos ejemplos de IDEs para PHP son PhpStorm o Visual Studio Code.

Además, si se desea trabajar con bases de datos, se necesitará tener instalado un gestor de bases de datos, como MySQL o MariaDB, y configurar PHP para trabajar con el gestor de base de datos elegido.

Instalación de entornos locales que soportan PHP

Hay varios paquetes de instalación de entornos locales que incluyen PHP y todas las herramientas necesarias para el desarrollo de aplicaciones web:

- **XAMPP:**
Es un paquete de instalación de código abierto que incluye Apache, PHP, MySQL, y otros componentes necesarios para el desarrollo de aplicaciones web. Está disponible para Windows, macOS y Linux.
- **WAMP:**
Es un paquete de instalación para Windows que incluye Apache, PHP y MySQL.
- **LAMP:**
Es un paquete de instalación para sistemas operativos basados en Unix, como Linux, que incluye Apache, PHP y MySQL.
- **MAMP:**
Es un paquete de instalación para macOS que incluye Apache, PHP y MySQL.
- **Vagrant:**
Es una herramienta para la creación y gestión de entornos virtuales que incluye la posibilidad de instalar y configurar Apache, PHP, MySQL y otros componentes necesarios para el desarrollo de aplicaciones web.

Estos paquetes de instalación son muy útiles para ahorrar tiempo y esfuerzo en la configuración de un entorno de desarrollo local, ya que incluyen todos los componentes necesarios y la configuración necesaria para poner en marcha un servidor web local en poco tiempo.

Editores online que soportan PHP

Otra opción al momento de programar en PHP es buscar editores en línea que soportan este lenguaje de programación, en Internet existen varios, y algunos de estos son:

- **CodePen:** Una plataforma en línea para compartir y probar pequeños fragmentos de código. Tiene una amplia comunidad de usuarios y permite la colaboración en tiempo real.
- **Repl.it:** Un editor de código en línea que soporta una amplia variedad de lenguajes, incluido PHP. Ofrece un entorno de desarrollo en línea intuitivo y fácil de usar.
- **JSFiddle:** Una herramienta en línea para probar y compartir fragmentos de código JavaScript, HTML y CSS. Permite la ejecución en tiempo real y la colaboración con otros usuarios.
- **PHPFiddle:** Un editor en línea especializado en PHP con soporte para diversas versiones del lenguaje. Ofrece una interfaz intuitiva y una variedad de características para ayudar a los desarrolladores a escribir y probar su código.
- **GitHub Codespaces:** Un editor de código en línea que permite a los usuarios trabajar en proyectos de forma colaborativa. Soporta PHP y una amplia variedad de otros lenguajes.
- **OnlineGDB:** Un editor en línea que permite a los usuarios escribir, compilar y ejecutar su código en un entorno de desarrollo en línea. Además de PHP, también soporta otros lenguajes como C, C++, Java y más.
- **Bitbucket Cloud:** Una plataforma en línea para alojar y colaborar en proyectos de software. Ofrece un editor en línea integrado para PHP y otros lenguajes.

Estos son solo algunos ejemplos de los mejores editores de PHP en línea gratuitos disponibles. Hay muchas otras opciones disponibles, por lo que es importante investigar y elegir el que mejor se adapte a sus necesidades y preferencias.

Sintaxis Básica PHP

La sintaxis básica de PHP incluye los siguientes componentes:

1. Etiquetas de apertura y cierre: el código PHP se escribe entre las etiquetas `<?php` y `?>`.

```
<?php

    // Código PHP aquí

?>
```

2. **Comentarios:** se pueden agregar comentarios con `//` para una sola línea o con `/*` y `*/` para múltiples líneas.

```
<?php

    // Este es un comentario de una sola línea

    /*
    Este es un comentario de
    múltiples líneas
    */

?>
```

3. **Variables:** se declaran con el signo `$` seguido por el nombre de la variable.

```
<?php

    // Declaración de una variable
    $nombre = "John Doe";
    $edad = 30;
    $salario = 3500.50;

?>
```


- 4. Sentencias:** se utilizan para controlar el flujo del programa, como if, else, for, while, etc.

```
<?php

    $edad = 30;

    // Sentencia if-else
    if($edad >= 18){
        echo "Eres mayor de edad";
    }
    else{
        echo "Eres menor de edad";
    }

    // Sentencia for
    for($i = 0; $i < 5; $i++){
        echo "Iteración " . $i . "<br>";
    }

    // Sentencia while
    $contador = 0;
    while($contador < 5){
        echo "Iteración " . $contador . "<br>";
        $contador++;
    }

?>
```

- 5. Funciones:** son bloques de código que se pueden llamar para realizar una tarea específica.

```
<?php

// Definición de una función
function saludar($nombre){
    echo "Hola, " . $nombre;
}

// Llamada a la función
saludar("John Doe");

?>
```

- 6. Operadores:** se utilizan para realizar operaciones matemáticas y comparaciones, como +, -, *, /, ==, etc.

```
<?php

// Asignación
$a = 10;
$b = 5;
$a = $b;
echo $a; // 5

// Suma
echo $a + $b; // 15

// Resta
echo $a - $b; // 5

// Multiplicación
echo $a * $b; // 50

// División
echo $a / $b; // 2

// Módulo (resto de la división)
echo $a % $b; // 0

// Exponenciación
echo $a ** $b; // 100000

// Igual a
echo $a == $b; // false
```

```
// No igual a
echo $a != $b; // true

// Mayor que
echo $a > $b; // true

// Menor que
echo $a < $b; // false

// Mayor o igual que
echo $a >= $b; // true

// Menor o igual que
echo $a <= $b; // false

// Idéntico
echo $a === $b; // false

// No idéntico
echo $a !== $b; // true

?>
```

Comentarios

Los comentarios en PHP son líneas de texto que no son interpretadas como código. Son utilizados para agregar explicaciones y notas dentro del código para facilitar su comprensión y mantenimiento.

Hay dos tipos de comentarios en PHP:

1. Comentarios de una línea:

Son representados por dos barras diagonales (//) al comienzo de una línea, o por el símbolo de gato (#).

2. Comentarios de varias líneas:

Son representados por /* y */.

La finalidad de los comentarios es mejorar la legibilidad y documentación del código, facilitando su mantenimiento y actualización.

```
<?php

// Este es un comentario de una única línea.
# Este es otro comentario de una única línea.

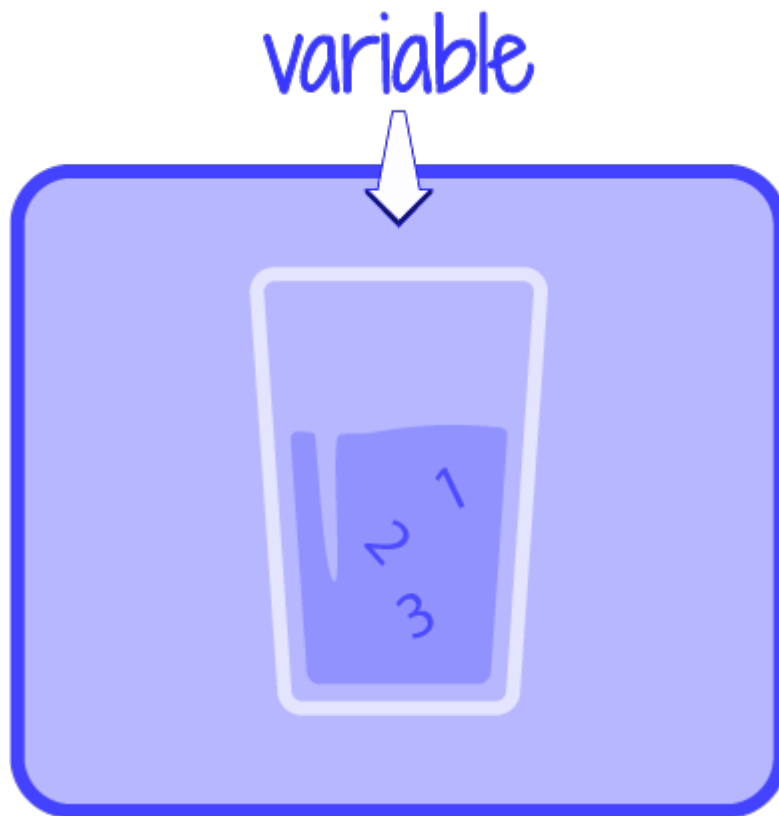
/*
Este es un
comentario
de múltiples líneas.
*/

?>
```

Variables

Las variables en PHP son espacios de almacenamiento que pueden contener valores de diferentes tipos, como números, cadenas de texto, arreglos, objetos, etc.

Pensá en las variables como contenedores para almacenar datos.



Imágenes extraídas de la web

En PHP, el nombre de una variable debe comenzar con un signo de dólar (\$), seguido por una o más letras, números o guiones bajos.

Una variable puede tener un nombre corto como "x" e "y" o un nombre más descriptivo como "edad", "modelo_auto", "nombre", "apellido", "total", etc).

Las reglas de las variables PHP son las siguientes:

- Una variable comienza con el signo dólar (\$), seguido del nombre de la variable.
- Una variable debe comenzar con una letra o guión bajo.
- Una variable no puede comenzar con un número.
- Una variable solo puede contener caracteres alfanuméricos y guión bajo (A-z, 0-9, y _).
- Los nombres de las variables son sensibles a mayúsculas (\$edad y \$EDAD son dos variables diferentes).

Aquí, hemos creado tres variables: \$nombre, \$edad y \$sueldo, y le hemos asignado valores.

Por ejemplo:

```
<?php

$nombre = "Juan";
$edad = 30;
$sueldo = 1500.50;

?>
```

El valor de una variable puede ser modificado en cualquier momento durante la ejecución de un script.

Por ejemplo:

```
<?php

$nombre = "Juan";
$edad = 30;

$nombre = "María";
$edad = $edad + 5;

echo $nombre; // Salida: María
echo $edad; // Salida: 35

?>
```


Además de los tipos básicos como números y cadenas, PHP también permite trabajar con arreglos y objetos. Un arreglo es una colección de valores indexados o asociativos.

Por ejemplo:

```
<?php

$colores = array("rojo", "verde", "azul");

echo $colores[0]; // Salida: rojo;

?>
```

Un objeto es una instancia de una clase y puede tener propiedades y métodos.

Por ejemplo:

```
<?php

class Coche{
    public $marca;
    public $modelo;

    function arrancar() {
        echo "El coche está arrancando.";
    }
}

$miCoche = new Coche();
$miCoche->marca = "Toyota";
$miCoche->modelo = "Corolla";

echo $miCoche->marca; // Salida: Toyota
$miCoche->arrancar(); // Salida: El coche está arrancando.

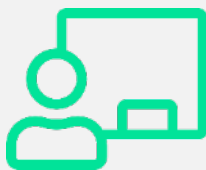
?>
```

En resumen, las variables son una parte fundamental del lenguaje PHP y son ampliamente utilizadas para almacenar y manipular datos.



Hemos llegado así al final de esta clase en la que vimos:

- Introducción a PHP: breve historia.
- ¿Qué puede hacerse con PHP?
- Sitios estáticos vs sitios dinámicos.
- Lenguajes de programación del lado del servidor.
- Instalación de PHP (instalación independiente).
- Instalación de entornos locales que soportan PHP.
- Editores online que soportan PHP.
- Sintaxis Básica PHP.
- Comentarios.
- Variables.



Te esperamos en la **clase en vivo** de esta semana.
No olvides realizar el **desafío semanal**.

¡Hasta la próxima clase!

Bibliografía

Documentación Oficial de PHP:

<https://www.php.net/manual/es/index.php>

W3Schools: PHP Tutorial:

<https://www.w3schools.com/php/default.asp>

Cabezas Granado, L., González Lozano, F., (2018). Desarrollo web con PHP y MySQL. Anaya Multimedia.

Heurtel, O., (2016). Desarrolle un sitio web dinámico e interactivo. Eni Ediciones.

Welling, L., Thompson, L., (2017). Desarrollo Web con PHP y MySQL. Quinta Edición. Editorial Anaya.

Para ampliar la información

MDN: Introducción al lado Servidor:

https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Introduction

Guía de HTML:

<https://developer.mozilla.org/es/docs/Learn/HTML>

Todo sobre PHP:

<https://desarrolloweb.com/home/php#track22>

PHP: The Right Way:

<https://phptherightway.com/>

PHP Programming Language Tutorial - Full Course:

https://www.youtube.com/watch?v=OK_JCtrrv-c&t=6s

PHP Full Course for non-haters:

<https://www.youtube.com/watch?v=zZ6vybT1HQs>

CURSO de php desde cero:

<https://www.youtube.com/watch?v=nCB1gEkRZ1g>