

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**DESARROLLO DE
BACKEND**

Tema

**PRESENTACIÓN DE LA UNIDAD
DIDÁCTICA.
BREVE INTRODUCCIÓN SOBRE EL
LENGUAJE PHP.**

INTRODUCCIÓN AL LENGUAJE PHP

1. ¿Qué es PHP?

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de programación de código abierto, ampliamente utilizado para el desarrollo web. Se ejecuta del lado del servidor y permite crear sitios web dinámicos e interactivos. Fue creado en 1994 por Rasmus Lerdorf y, desde entonces, ha evolucionado hasta convertirse en uno de los pilares del desarrollo web moderno.

Su principal ventaja es la capacidad de integrarse fácilmente con HTML, lo que facilita la creación de páginas que combinan contenido estático y dinámico. Además, PHP es compatible con una gran variedad de servidores web y sistemas operativos.

2. Características Principales de PHP

Lenguaje interpretado: No requiere compilación, el servidor interpreta el código directamente.

Open Source: Es gratuito y cuenta con una amplia comunidad de desarrolladores que contribuyen a su mejora.

Multiplataforma: Funciona en sistemas como Windows, Linux, macOS y otros.

Integración con bases de datos: Soporta múltiples sistemas de gestión de bases de datos como MySQL, MariaDB, PostgreSQL, Oracle, entre otros.

Facilidad de aprendizaje: Su sintaxis es sencilla, similar a C, Java y Perl.

Gran biblioteca de funciones: Incluye funciones predefinidas para manejar cadenas, archivos, imágenes, fechas, sesiones y más.

Escalabilidad: Apto para proyectos pequeños y grandes aplicaciones empresariales.

3. Instalación y Entorno de Trabajo

Para trabajar con PHP es necesario contar con un entorno de servidor. Algunas opciones populares son:

XAMPP: Paquete que incluye Apache, MySQL/MariaDB, PHP y phpMyAdmin.

WAMP (Windows), **MAMP** (Mac) o **LAMP** (Linux).

Servidores en la nube: Como AWS, Azure o Google Cloud.

Docker: Contenedores para entornos de desarrollo aislados.

La instalación básica incluye:

Un servidor web (Apache, Nginx, IIS).

El motor de PHP.

Un sistema de base de datos (generalmente MySQL o MariaDB).

4. Sintaxis Básica de PHP

El código PHP se incrusta dentro de archivos HTML utilizando etiquetas especiales:

```
php

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

- Comentarios:

```
php

// Comentario de una línea
# Otra forma de comentar
/* Comentario de varias líneas */
```

- Variables: Siempre inician con \$.

```
php

$nombre = "Juan";
$edad = 25;
```

Tipos de datos: String, Integer, Float, Boolean, Array, Object, NULL.

5. Uso de PHP en Desarrollo Web

- PHP se utiliza principalmente para:
- Generar contenido dinámico.
- Procesar formularios HTML y validar datos.
- Conectar y manipular bases de datos.
- Gestionar sesiones y cookies.
- Consumir APIs.
- Procesar archivos (lectura, escritura, subida).
- Desarrollar aplicaciones completas (usando frameworks como Laravel, Symfony, CodeIgniter).

6. Integración con Bases de Datos

La combinación de PHP y MySQL es una de las más utilizadas en el desarrollo web.

Ejemplo básico de conexión con PDO:

```
php

<?php
try {
    $conexion = new PDO("mysql:host=localhost;dbname=mi_bd", "usuario", "contraseña");
    $conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Conexión exitosa";
} catch (PDOException $e) {
    echo "Error: " . $e->getMessage();
}
?>
```

7. Ventajas y Desventajas de PHP

Ventajas:

- Amplia comunidad y documentación.
- Fácil integración con HTML y bases de datos.
- Curva de aprendizaje corta.
- Compatible con la mayoría de los servidores.

Desventajas:

- Código desordenado si no se aplican buenas prácticas.
- Mayor seguridad requerida si no se validan entradas.
- No es el más rápido en ejecución comparado con algunos lenguajes modernos.

8. Frameworks y Herramientas Populares

- Laravel: Enfoque moderno, arquitectura MVC, seguridad y herramientas integradas.
- Symfony: Robusto y modular.
- CodeIgniter: Ligero y fácil de configurar.
- WordPress: CMS basado en PHP.
- Composer: Gestor de dependencias de PHP.

9. Buenas Prácticas

- Separar la lógica de negocio de la presentación (MVC).
- Validar y filtrar todos los datos de entrada.
- Usar sentencias preparadas en bases de datos para prevenir inyecciones SQL.
- Mantener el código comentado y organizado.
- Actualizar versiones de PHP y librerías.

10. Conclusión

PHP sigue siendo una herramienta fundamental en el desarrollo web. Aunque han surgido otros lenguajes y tecnologías, su flexibilidad, comunidad y constante evolución lo mantienen vigente. Con buenas prácticas y el uso de frameworks modernos, PHP permite desarrollar desde pequeños sitios hasta sistemas empresariales complejos.

