

# R1.1 - Plataformas de Programación Web en Entorno Servidor

## 1. Introducción a la programación web

- **Páginas web:**
  - **Estáticas:** Contenido fijo, creado con HTML/CSS.  
**Ventajas:** fácil creación, buen SEO.  
**Desventajas:** actualización manual y poco flexibles.
  - **Dinámicas:** Generadas en tiempo real por el servidor o el cliente.  
Lenguajes como PHP, ASP, JSP o JavaScript permiten crear estas páginas.
- **Aplicaciones web:** Ejemplos como correo electrónico, editores online.  
**Ventajas:**
  - No requieren instalación.
  - Multiplataforma (funcionan en cualquier navegador).
  - Centralización en el servidor.
  - Accesibles desde cualquier lugar con conexión.
  - **Desventajas:** velocidad dependiente del servidor, interfaz limitada al navegador, y necesidad de conexión a internet.

## 2. Ejecución de código

- En páginas dinámicas, el código se ejecuta en el servidor y se traduce a HTML para enviarse al navegador.
- **Comparativa PHP vs. JavaScript:**
  - **PHP (Servidor):** Se ejecuta en el servidor, velocidad media-alta, alta seguridad, ideal para backend.
  - **JavaScript (Cliente):** Se ejecuta en el navegador, velocidad alta, seguridad media, excelente para interfaces interactivas.

## 3. Tecnologías del entorno servidor

- **Componentes principales:**
  - **Servidor web:** Atiende peticiones y envía respuestas.
  - **Módulo de ejecución:** Interpreta código (ej., PHP).
  - **Base de datos:** Almacena y recupera datos.
  - **Lenguaje de programación:** Desarrolla la lógica.
- **Organización en capas:**
  - **Capa cliente:** Presentación (HTML, CSS, JavaScript).
  - **Capa lógica:** Procesos (PHP, Python).
  - **Capa de acceso a datos:** Gestión de datos en bases de datos.

## 4. Arquitecturas y plataformas

- **LAMP/WAMP/MAMP:** Linux/Windows/Mac, Apache, MySQL, PHP. Open source, ideal para proyectos medianos.
- **Java EE:** Complejo, para proyectos grandes.
- **ASP.NET:** Comercial, Microsoft SQL Server y IIS.
- **CGI/Perl:** Antigua y menos eficiente.

## 5. Lenguajes de programación

- **Interpretados:** PHP, Python. Portables pero más lentos.
  - **Compilados:** C#, Java. Rápidos pero menos portables.
  - **Embebidos:** Código HTML que incluye scripts PHP o viceversa.
- 

# Resumen del Documento R1.2 - Programación PHP

## 1. Introducción a PHP

- **PHP (PHP Hypertext Preprocessor):** Lenguaje interpretado para crear páginas dinámicas.
- **Sintaxis similar a C/C++.**
- **Delimitadores:** `<?php ... ?>`.

## 2. Inserción de código en HTML

- Código PHP puede incluirse en cualquier parte del HTML.
- La ejecución reemplaza el script por su resultado.
- **Comentarios:**
  - Una línea: `//` o `#`.
  - Varias líneas: `/* comentario */`.

## 3. Uso básico de PHP

- **Impresión:**
  - `echo` y `print` imprimen texto o variables.
  - Uso de comillas dobles (") o simples (').
  - Caracteres especiales deben escaparse con `\`.
- **Variables:**
  - Se identifican con `$` (ejemplo: `$nombre`).
  - No necesitan declaración previa.
  - PHP es **sensible a mayúsculas y minúsculas**.

## 4. Tipos de datos

- **Escalares:** `boolean`, `integer`, `float`, `string`.
- **Compuestos:** `array`, `object`.

- **Especiales:** `NULL`, `resource`.
- **Conversión de tipos:**
  - Automática: `"20" + 15 = 35`.
  - Manual: `(int)$var`, `(string)$var`.

## 5. Operadores y operaciones

- **Aritméticos:** `+`, `-`, `*`, `/`, `%`.
- **Asignación:** `=`, `+=`, `-=`, `*=`.
- **Concatenación:** `.`, ejemplo: `"Hola " . "Mundo"`.
- **Incremento/decremento:** `++`, `--`.

## 6. Funciones útiles

- Comprobación del tipo de una variable:
  - `is_int()`, `is_bool()`, `is_string()`, `is_array()`, `is_object()`.
- Estado de una variable:
  - `isset()` verifica si tiene un valor.
  - `empty()` verifica si está vacía.
  - `unset()` borra el valor de una variable.

## 7. Ámbito de las variables

- **Local:** Accesible solo dentro de funciones.
- **Global:** Declaradas fuera de funciones, accesibles con `global`.