



DESARROLLO WEB

USO DE SERVIDORES EN SITIOS DINÁMICOS: PHP, NODE.JS Y PYTHON

USO DE SERVIDORES EN SITIOS DINÁMICOS: PHP, NODE.JS Y PYTHON

Los servidores en sitios dinámicos ejecutan código en tiempo real para procesar solicitudes, generar contenido personalizado y gestionar la lógica de negocio de las aplicaciones web.



PHP ha sido tradicionalmente una de las tecnologías más utilizadas para desarrollo web del lado del servidor, especialmente en aplicaciones que requieren integración estrecha con bases de datos relacionales y despliegue en servidores web tradicionales como Apache. PHP ofrece una sintaxis accesible para desarrolladores que comienzan en el desarrollo web, con un amplio ecosistema de frameworks como Laravel y Symfony que proporcionan herramientas

robustas para el desarrollo de aplicaciones empresariales. La integración nativa con bases de datos MySQL y la disponibilidad de hosting económico han contribuido a su popularidad en proyectos de pequeña y mediana escala.

Vara Mesa, Verde Marín y López Sanz (2015) describen cómo Node.js ha revolucionado el desarrollo web al permitir el uso de JavaScript tanto en el cliente como en el servidor, facilitando el intercambio de código y conocimiento entre el desarrollo frontend y backend. Se implementa un modelo de programación asíncrona basado en eventos que resulta especialmente eficiente para aplicaciones con alta concurrencia y operaciones de E/S intensivas. Python, con frameworks como Django y Flask, se ha establecido como una opción popular para aplicaciones web que requieren integración con sistemas de análisis de datos, inteligencia artificial, o procesamiento científico, aprovechando el rico ecosistema de bibliotecas disponibles en el lenguaje.

En proyectos de ingeniería de software, se observa el uso de estas tecnologías para desarrollar herramientas especializadas: PHP para sistemas de gestión de contenido y portales corporativos, Node.js para aplicaciones en tiempo real como sistemas de chat o dashboards colaborativos, y Python para herramientas de análisis de código, sistemas de CI/CD, y aplicaciones que requieren procesamiento de datos complejos. La elección de la tecnología del servidor afecta significativamente la arquitectura del sistema, las posibilidades de escalabilidad, y la integración con otras herramientas del ecosistema de desarrollo.

Para ilustrar las diferencias entre estas tecnologías de servidor, se propone crear tres versiones simples de una API REST que gestione información de proyectos de software. Se implementa en PHP una versión que utilice arrays asociativos para simular una base de datos, mostrando la sintaxis directa del lenguaje para procesamiento de datos. Se desarrolla en Node.js una versión que utilice Express.js y módulos ES6, demostrando la programación asíncrona y el manejo de JSON nativo. Se crea en Python una versión utilizando Flask que muestre la simplicidad del lenguaje y su capacidad para manejar datos estructurados. El resultado permite comparar directamente las diferencias sintácticas, paradigmas de programación y capacidades específicas de cada tecnología, proporcionando criterios objetivos para la selección de la tecnología más apropiada, según el contexto del proyecto.