

# Comparativa de lenguajes y frameworks

En esta práctica investigaremos sobre las tecnologías de programación web en entorno servidor. Para explicarlas, analizaremos sus características, ventajas, frameworks asociados y casos de uso recomendados.

## **PHP**

### **Características del lenguaje:**

PHP es un lenguaje de programación de código abierto que se utiliza en gran parte para crear sitios web y aplicaciones web, es un lenguaje de propósito general que se ejecuta en la parte del servidor. Además de ser un lenguaje interpretado, tiene varias formas de uso; con scripts, de forma estructurado o en programación de objetos. También soporta la mayoría de bases de datos, ya sea MySQL, PostgreSQL, SQL Server, MongoDB...

### **Ventajas e inconvenientes:**

PHP presenta ventajas sobre otros lenguajes. Al ser un lenguaje de código abierto, cualquier programador tiene acceso a él sin costos. La curva de aprendizaje es baja y comprensible y se simplifica mucho gracias a la gran cantidad de tutoriales. Además, de tener una fácil instalación y soportar una gran cantidad de datos.

Sin embargo, PHP no dispone de las herramientas de depuración necesarias para buscar errores. Otro inconveniente que destacar es la baja seguridad, para garantizar la seguridad de los datos es necesario tener buenos conocimientos técnicos o trabajar con SQL.

### **Frameworks disponibles:**

PHP dispone de un gran abanico de frameworks que podemos utilizar, entre los que podemos destacar Laravel, que es fácil de entender y sencillo para trabajar; CodeIgniter, que usa una arquitectura de MVC o Symfony, que cuenta con un sistema de componentes que permite elegir las funciones de PHP que necesites o usar todo el framework.

### **Situaciones o casos de uso recomendados:**

La mejor situación para usar PHP y exprimir al máximo sus funciones sería para crear sitios web estáticos y dinámicos. Los sitios estáticos pueden usar PHP incluyendo fragmentos de HTML y los sitios dinámicos generan contenido de

manera personalizada para el usuario en función de sus acciones o preferencias.

## **Node.js**

### **Características del lenguaje:**

Node.js es un entorno de ejecución que se utiliza para ejecutar JavaScript fuera del navegador. Este entorno es fácil de entender y con muchos tutoriales para aprender. También, existe un amplio conjunto de paquetes de código abierto para ayudarte en tu trabajo y permite crear sitios web SaaS, aplicaciones de escritorio e incluso aplicaciones móviles.

### **Ventajas e inconvenientes:**

Node.js ofrece ventajas como su eficiencia usando miles de conexiones simultáneas con un solo hilo, una gran cantidad de paquetes disponibles gracias a npm o una de las comunidades más grandes del desarrollo web.

Por otro lado, opera solo en un hilo, lo que no lo hace muy buena opción para tareas de CPU intensivas.

### **Frameworks disponibles:**

Los frameworks más utilizados de Node.js son los siguientes. Express, se utiliza bastante para entornos serverless. Fastify, es uno de los frameworks más importantes de Node.js con actualizaciones constantes y una comunidad activa. Nestjs, con este framework se puede crear aplicaciones backend completas con un diseño modular.

### **Situaciones o casos de uso recomendados:**

Se debe usar Node.js en caso de querer crear una aplicación web con alta concurrencia, sistemas en tiempo real o microservicios. Especialmente si se domina JavaScript y necesitas un entorno de desarrollo rápido y escalable.

## **Python**

### **Características del lenguaje:**

Python es un lenguaje de programación de alto nivel, interpretado y de código abierto que puede utilizarse para multitud de usos, desde desarrollo web y software hasta ciencia de datos. Tiene bibliotecas extensas que facilitan la

realización de tareas complejas. Presenta una fácil sencilla sintaxis y al ejecutarse línea por línea facilita la depuración de errores.

### **Ventajas e inconvenientes:**

Usar Python ofrece varias ventajas, de todas ellas podemos destacar las siguientes. Es muy utilizado en muchas industrias lo que aumenta las oportunidades laborales de los programadores. Sintaxis simple y clara que reduce la curva de aprendizaje.

Desafortunadamente, Python cuenta algunos inconvenientes. Algunos de ellos son los problemas de compresión asíncrona, puede presentar menos velocidad de ejecución que otros lenguajes y puede tener problemas de integración con C y C++.

### **Frameworks disponibles:**

Python dispone de 3 tipos de frameworks: Full-Stack, Microframeworks y Asíncronos. Algunos de ellos es Flask, CherryPy o Django.

### **Situaciones o casos de uso recomendados:**

Las situaciones más comunes de uso de Python son para el desarrollo de sitios web y software, la automatización de tareas y el análisis y visualización de datos.

## **Java**

### **Características del lenguaje:**

Java es un lenguaje de programación de propósito general orientado a objetos. Este lenguaje permite estructurar el código en objetos reutilizables y gracias a JVM puede correr en diferentes sistemas operativos sin necesidad de modificarlo. Además, tiene multihilo y un lenguaje robusto que detecta muchos errores en tiempo de compilación.

### **Ventajas e inconvenientes:**

Java tiene puntos fuertes como su independencia de la plataforma, una sólida seguridad, escalabilidad y muchas herramientas para utilizar.

Lo negativo de Java es la sobrecarga de JVM, su mayor uso de memoria y los tiempos de arranque más lentos en comparación con otros lenguajes.

### **Frameworks disponibles:**

Spring es conocido por su amplia gama de características y módulos que facilitan el desarrollo de aplicaciones. Mientras que, Hibernate por ejemplo, es un framework de Java que se utiliza para el mapeo objeto-relacional y simplifica la interacción entre las aplicaciones Java y las bases de datos relacionales.

### **Situaciones o casos de uso recomendados:**

Java es ideal para backends complejos, software empresarial, aplicaciones Android, microservicios o cualquier proyecto que necesite estabilidad a largo plazo e integración tecnológica avanzada.

## **Conclusión**

Personalmente, utilizaría Node.js con Express. Con este podría utilizar el mismo lenguaje en frontend y backend, reduciendo la complejidad y mejorando la comprensión. Además, cuenta con una comunidad muy activa con muchos tutoriales. También es muy eficiente para manejar conexiones simultáneas gracias a su modelo basado en eventos y usos de un solo hilo y su fácil integración con bases de datos NoSQL y SQL.