

**Entornos Tecnológicos y Frameworks de Desarrollo
(Java, .NET Core, PHP, Python, Node.js)**

Alumno:

Olivas Beltran Luis Antonio

Curso:

Desarrollo web lado del servidor

Profesor:

Cazarez Alderete José Manuel

Escuela:

Facultad de informática culiacan UAS

Grupo:

2-1

ÍNDICE

1. Introducción

Java

- Historia y evolución
- Entorno de desarrollo
- Frameworks populares
- Usos actuales0

2. .NET Core

- Historia y características
- Ventajas y arquitectura
- Frameworks asociados
- Aplicaciones prácticas

3. PHP

- Origen y evolución
- Ventajas y desventajas
- Frameworks más usados
- Casos de uso

4. Python

- Breve historia
- Características destacadas
- Frameworks relevantes
- Aplicaciones actuales

5. Node.js

- Qué es y cómo funciona
- Ventajas y arquitectura
- Frameworks y librerías populares
- Usos comunes

Comparativa general

6. Conclusión

Bibliografía

Introducción

El desarrollo de software moderno se apoya en diversos entornos tecnológicos que permiten construir aplicaciones eficientes, escalables y mantenibles. Este informe ofrece una visión general de cinco tecnologías ampliamente utilizadas: Java, .NET Core, PHP, Python y Node.js. Cada uno tiene características distintivas, fortalezas y casos de uso específicos.

Java

Historia y evolución

Java fue desarrollado por Sun Microsystems en 1995 y actualmente es mantenido por Oracle. Se diseñó con la filosofía "escribe una vez, ejecuta en cualquier lugar".

Entorno de desarrollo

Utiliza la Java Virtual Machine (JVM), que permite ejecutar el código en múltiples plataformas. IDEs comunes incluyen IntelliJ IDEA, Eclipse y NetBeans.

Frameworks populares

- **Spring Framework:** Muy usado en aplicaciones empresariales.
- **Hibernate:** ORM para persistencia de datos.
- **Struts:** Para desarrollo de aplicaciones web MVC.

Usos actuales

Java es ampliamente utilizado en aplicaciones empresariales, Android, sistemas bancarios y servidores backend.

.NET Core

Historia y características

.NET Core es un framework de desarrollo multiplataforma de código abierto creado por Microsoft en 2016, sucesor moderno de .NET Framework.

Ventajas y arquitectura

- Multiplataforma (Windows, Linux, macOS)
- Alto rendimiento
- Soporte para contenedores

Frameworks asociados

- **ASP.NET Core:** Para aplicaciones web y APIs REST.
- **Entity Framework Core:** ORM para bases de datos.
- **Blazor:** Desarrollo web con C# en lugar de JavaScript.

Aplicaciones prácticas

Se utiliza en aplicaciones web, microservicios, APIs y software de escritorio multiplataforma

PHP

Origen y evolución

PHP fue creado por Rasmus Lerdorf en 1995. Es un lenguaje de scripting muy popular para desarrollo web.

Ventajas y desventajas

- **Ventajas:** Fácil de aprender, gran comunidad, integración con bases de datos.
- **Desventajas:** Problemas de seguridad si no se usa correctamente, sintaxis inconsistente.

Frameworks más usados

- **Laravel:** El más popular actualmente.
- **Symfony:** Muy estructurado y flexible.
- **CodeIgniter:** Ligero y fácil de configurar.

Casos de uso

Amplio uso en desarrollo de sitios web, plataformas CMS (WordPress, Joomla) y e-commerce.

Python

Breve historia

Python fue creado por Guido van Rossum en 1991. Se enfoca en la legibilidad del código y la productividad del desarrollador.

Características destacadas

- Sintaxis clara y concisa
- Tipado dinámico
- Gran cantidad de bibliotecas

Frameworks relevantes

- **Django**: Completo y seguro.
- **Flask**: Minimalista y flexible.
- **FastAPI**: Moderno y muy rápido para APIs.

Aplicaciones actuales

Se usa en desarrollo web, ciencia de datos, automatización, inteligencia artificial y scripting.

Node.js

Qué es y cómo funciona

Node.js es un entorno de ejecución para JavaScript del lado del servidor, basado en el motor V8 de Chrome. Permite construir aplicaciones rápidas y escalables.

Ventajas y arquitectura

- No bloqueante (event-driven)
- Gran rendimiento para aplicaciones I/O intensivas
- Gran ecosistema de paquetes (NPM)

Frameworks y librerías populares

- **Express.js**: Minimalista y flexible.
- **NestJS**: Arquitectura modular basada en TypeScript.
- **Socket.IO**: Para comunicación en tiempo real.

Usos comunes

Aplicaciones web en tiempo real, APIs RESTful, microservicios, aplicaciones serverless.

Comparativa general

Tecnología	Lenguaje Base	Multiplataforma	Framework Principal	Uso Común
Java	Java	Sí	Spring	Backend empresarial, Android
.NET Core	C#	Sí	ASP.NET Core	Web, APIs, escritorio
PHP	PHP	Sí	Laravel	Web (CMS, tiendas online)
Python	Python	Sí	Django, Flask	Web, ciencia de datos, IA
Node.js	JavaScript	Sí	Express, NestJS	Web en tiempo real, APIs

Conclusión

Cada entorno tecnológico tiene ventajas específicas y es más adecuado para ciertos tipos de proyectos. Java y .NET Core dominan en el sector empresarial, PHP sigue fuerte en el desarrollo web, Python lidera en inteligencia artificial y Node.js es ideal para aplicaciones modernas de alto rendimiento. La elección adecuada depende de los requisitos del proyecto, el equipo de desarrollo y el ecosistema tecnológico existente.

Bibliografía

- Oracle: <https://www.oracle.com/java/>
- Microsoft: <https://dotnet.microsoft.com/>
- PHP Manual: <https://www.php.net/>
- Python.org: <https://www.python.org/>
- Node.js: <https://nodejs.org/>
- Laravel: <https://laravel.com/>
- Django: <https://www.djangoproject.com/>
- Express: <https://expressjs.com/>