¿Qué es Rust?

Rust es un lenguaje de sistemas moderno, diseñado para garantizar seguridad de memoria sin recolector de basura y rendimiento comparable a C y C++.

Su desarrollo comenzó en 2006 por Graydon Hoare y fue patrocinado por Mozilla, culminando con su primera versión estable en mayo de 2015.

Rust resuelve problemas clásicos de lenguajes de bajo nivel como punteros nulos, doble liberación y condiciones de carrera data races, mediante un sistema de propiedad Ownership y un verificador de préstamos Borrowing que el compilador valida en tiempo de compilación.

Rust no solo permite crear aplicaciones completas desde cero, puesto que su diseño de abstractions y memory safety lo convierte en una opción atractiva para reescribir componentes críticos en ecosistemas ya existentes.

Ejemplos concretos:

1. Firefox sustituyó partes de su motor CSS con Servo, reduciendo errores de memoria en un 70 %.
2. Discord migró su servicio de voz a Rust y dividió por 10 el uso de CPU frente a la versión en Go.
3. Dropbox reescribió su sistema de sincronización con Rust + Tokio, logrando menos latencia y menor consumo de memoria sin aumentar el coste de hardware.

Estas migraciones demuestran que Rust puede aumentar el rendimiento y reducir costes operativos sin sacrificar seguridad ni productividad.

La seguridad garantizada por su diseño ha generado una confianza excepcional entre los desarrolladores. De manera ininterrumpida desde 2016, Rust ha sido votado como el “lenguaje de programación más querido” en la encuesta anual de Stack Overflow.

Esta popularidad y la alta satisfacción de los desarrolladores son un fuerte indicador de su productividad y baja curva de frustración una vez dominado.

El compilador de Rust puede parecer exigente al principio, pero está diseñado para detectar errores *antes* de que tu código se ejecute. Esto significa menos bugs en producción y noches más tranquilas.

Además, Rust:

- **Compila directamente a código máquina**: Sin overhead de runtime ni máquinas virtuales
- **Soporta múltiples hilos sin data races**: El compilador garantiza que tu código concurrente es seguro
- **Elimina categorías enteras de bugs**: Muchos errores de concurrencia simplemente no pueden ocurrir

Por qué las empresas eligen Rust

Las razones principales que impulsan esta adopción son:

1. **Menos bugs en producción**: El compilador atrapa errores antes del despliegue
2. **Mayor rendimiento**: Comparable a C/C++ sin sacrificar seguridad
3. **Menor uso de recursos**: Traduce en ahorro de costos de infraestructura
4. **Código más mantenible**: Las garantías del lenguaje facilitan refactorings grandes