

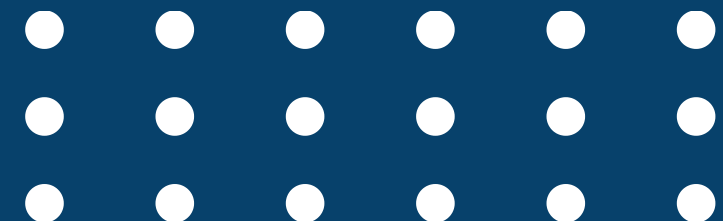
Comparativa de Lenguajes de Programación Emergentes para Aplicaciones en Infraestructura

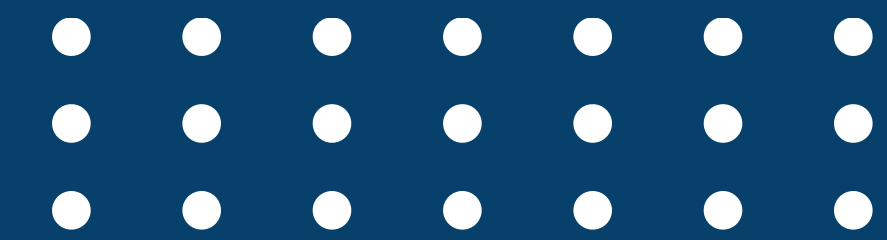
Por

- Enrique Pérez
 - Martin Zambonino
- 
- 

Índice

- 1 Introducción
- 2 Lenguajes emergentes
- 3 Optimización para Aplicaciones Industriales
- 4 Robustez y Fiabilidad
- 5 Escalabilidad
- 6 Compatibilidad y Ecosistema
- 7 Casos de Uso en Infraestructura
- 8 Rust
- 9 Casos de Uso en Rust
- 10 Python
- 11 Casos de Uso en Python
- 12 Comparación Directa
- 13 Conclusión





Introducción

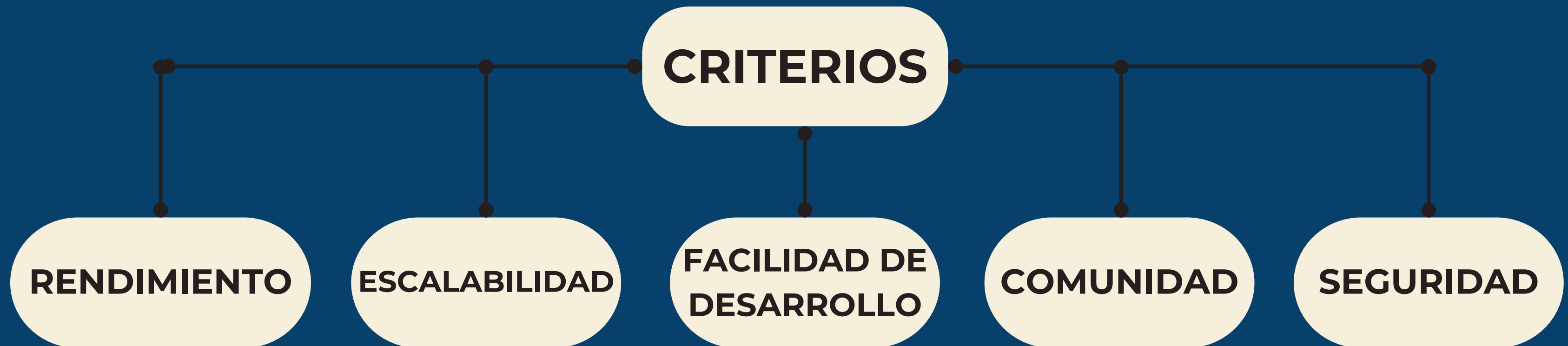
Analizaremos Rust y Python en la industria de infraestructura.

Analizaremos Rust y Python en la industria de infraestructura.



LENGUAJES EMERGENTES

NUEVOS Y POPULARES EN APLICACIONES
INDUSTRIALES



OPTIMIZACIÓN PARA APLICACIONES INDUSTRIALES

**DESEMPEÑO
Y EFICIENCIA**

CONCURRENCIA

Crucial para la industria

Múltiples tareas simultáneas

ROBUSTEZ Y FIABILIDAD

**TOLERANCIA A
FALLOS**

Función continua ante fallos

SEGURIDAD

Protección contra vulnerabilidades



COMPATIBILIDAD Y ECOSISTEMA

INTEROPERABILIDAD

Integración con sistemas
existentes

**LIBRERÍAS Y
FRAMEWORKS**

Disponibilidad para el desarrollo
industrial

ESCALABILIDAD

**ADAPTABILIDAD A
PROYECTOS
GRANDES**

Crecimiento sin pérdida de
rendimiento

MANTENIMIENTO

Facilidad de actualización

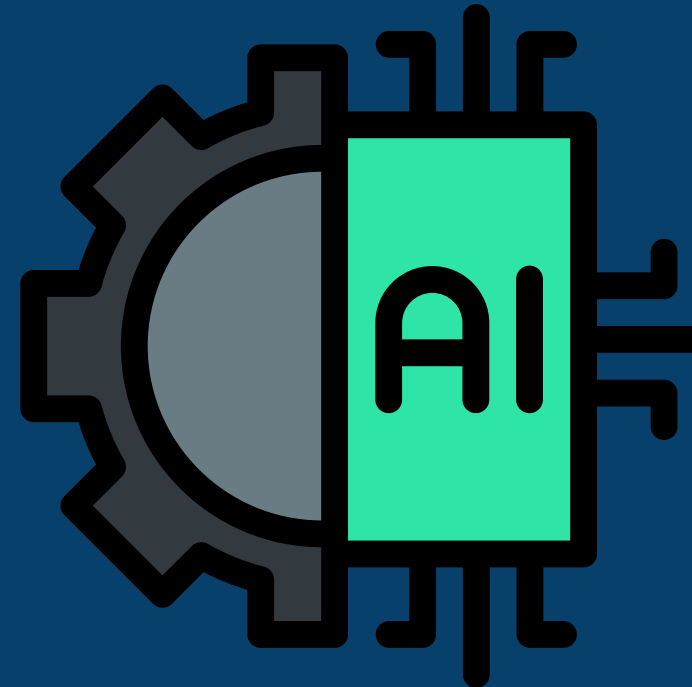


CASOS DE USO EN INFRAESTRUCTURA

**MONITOREO Y
CONTROL DE SISTEMAS**

**AUTOMATIZACIÓN DE
PROCESOS**

**GESTIÓN DE
RECURSOS**



RUST

- **Historia: Creado por Mozilla en 2010**
- **Descripción: Seguro y eficiente**
- **Ventajas:**
 - **Alta seguridad**
 - **Excelente rendimiento**
 - **Comunidad activa**
- **Desventajas:**
 - **Curva de aprendizaje**
 - **Ecosistema en desarrollo**

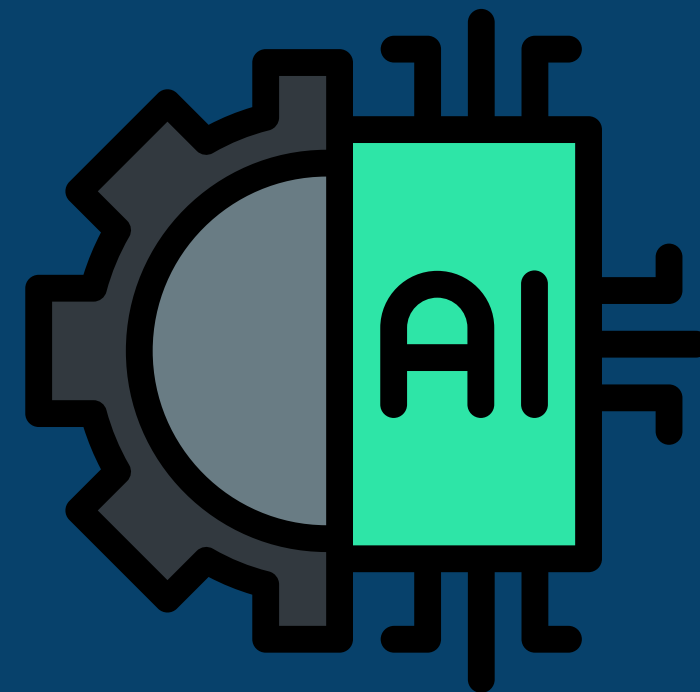


**The Rust
Programming
Language**

CASOS DE USO EN RUST

**MONITOREO Y
CONTROL DE
SISTEMAS CRÍTICOS**

**AUTOMATIZACIÓN
DE PROCESOS**



PYTHON

- **Historia: Creado por Guido van Rossum en 1991**
- **Descripción: Versátil y fácil de usar**
- **Ventajas:**
 - **Facilidad de uso**
 - **Amplia comunidad**
 - **Vasto ecosistema de librerías**
- **Desventajas:**
 - **Menor rendimiento**
 - **No ideal para tareas de tiempo real**

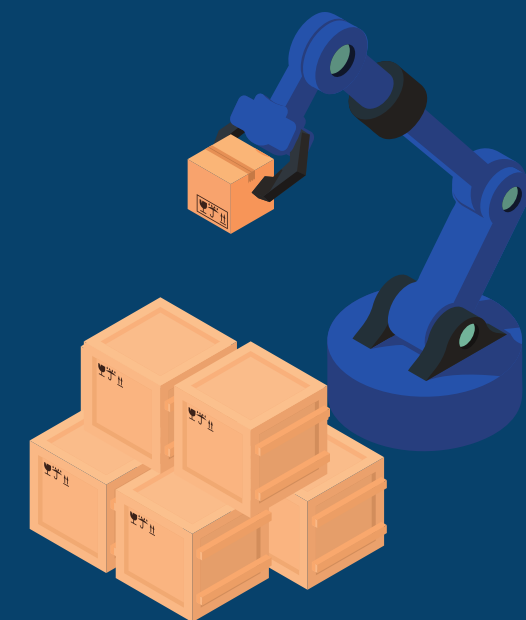
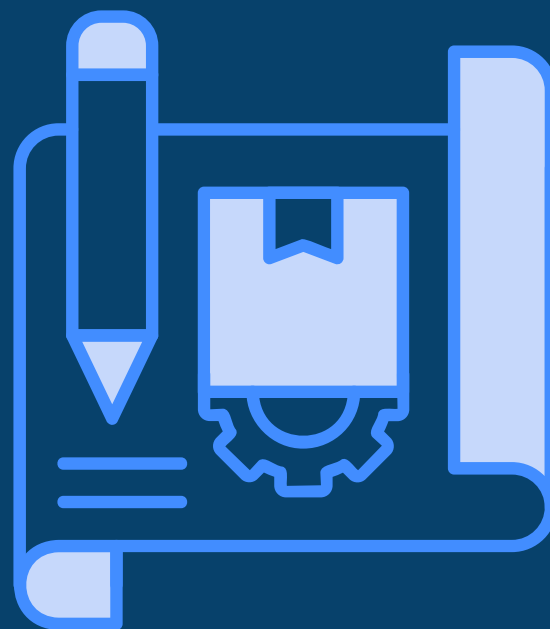


CASOS DE USO EN PYTHON

**DESARROLLO RÁPIDO
DE PROTOTIPOS**

**GESTIÓN DE
RECURSOS**

**SCRIPTS DE
AUTOMATIZACIÓN**



COMPARACIÓN DIRECTA

Característica	Rust	Python
Rendimiento	 Alto (ideal para tareas críticas y tiempo real)	 Moderado (adecuado para la mayoría de aplicaciones)
Escalabilidad	 Alta (eficiente en el uso de recursos)	 Alta (requiere herramientas y arquitecturas específicas)
Facilidad de Desarrollo	 Moderada (curva de aprendizaje más pronunciada)	 Alta (sintaxis sencilla y gran cantidad de recursos)
Soporte y Comunidad	 En crecimiento (comunidad activa y apasionada)	 Extenso (comunidad masiva y recursos abundantes)
Seguridad	 Muy alta (sistema de tipos y propiedad garantizan seguridad de memoria)	 Moderada (tipado dinámico puede llevar a errores en tiempo de ejecución)