

# Especificación de Requisitos de Software (ERS)

Sistema: Argos

---

## 1. Introducción

### 1.1 Propósito

El propósito de este documento es definir los requisitos del sistema Argos, una herramienta de supervisión de recursos del sistema escrita en Rust. Este documento está dirigido a desarrolladores, ingenieros de software, administradores de sistemas y cualquier parte interesada en el desarrollo, integración o uso de Argos.

### 1.2 Alcance

Argos es un sistema modular para la monitorización en tiempo real de procesos, permitiendo observar el uso de recursos como CPU, memoria y disco, así como establecer alertas, realizar logging y exportar datos. El sistema está diseñado para ser eficiente, seguro y reutilizable, con interfaces configurables y capacidad de integración en otros proyectos.

### 1.3 Definiciones, acrónimos y abreviaturas

- **ERS**: Especificación de Requisitos de Software
- **CLI**: Interfaz de Línea de Comandos
- **FFI**: Foreign Function Interface

### 1.4 Referencias

- IEEE 830-1998, IEEE Recommended Practice for Software Requirements Specifications

### 1.5 Visión general del documento

Este documento describe los objetivos, funcionalidades, restricciones y requisitos del sistema Argos.

---

## 2. Descripción General

### 2.1 Perspectiva del producto

Argos es una herramienta independiente y modular, inspirada en Argos Panoptes de la mitología griega, que proporciona vigilancia constante sobre procesos críticos. Puede funcionar como aplicación CLI, biblioteca embebida o integrarse en sistemas de monitoreo más complejos.

### 2.2 Funcionalidades del producto

- Monitoreo en tiempo real de uso de CPU, memoria y disco por proceso.
- Selección de procesos y configuración de intervalos de muestreo.
- Definición de umbrales de alerta.
- Logging y exportación de datos en formatos estándar (JSON, CSV).
- Arquitectura modular y extensible.
- Documentación y ejemplos prácticos.

### 2.3 Usuarios del producto

- Desarrolladores de software
- Ingenieros SRE / DevOps
- Docentes y estudiantes
- Sistemas de integración continua (CI)

### 2.4 Restricciones

- El sistema debe estar implementado en Rust.
- Debe ser multiplataforma (Windows, Linux, preferentemente MacOS).
- El rendimiento no debe afectar significativamente los procesos supervisados.

### 2.5 Suposiciones y dependencias

- El usuario tiene permisos suficientes para acceder a la información de los procesos.
- El entorno de ejecución soporta Rust y sus dependencias.

### 3. Requisitos Específicos

#### 3.1 Requisitos funcionales

- RF1: El sistema debe permitir seleccionar uno o varios procesos para su monitoreo.
- RF2: Debe capturar y mostrar en tiempo real el uso de CPU, memoria y disco de los procesos seleccionados.
- RF3: El usuario debe poder definir intervalos de muestreo y umbrales de alerta.
- RF4: El sistema debe generar logs y exportar datos en formatos JSON y CSV.
- RF5: Debe proveer una CLI y una biblioteca reutilizable (argos-core).
- RF6: Debe permitir la integración mediante FFI con otros lenguajes (Python, C).
- RF7: El sistema debe permitir comparar el historial de uso de recursos de un proceso entre diferentes ejecuciones, facilitando el análisis de optimización y evolución del proceso.

#### 3.2 Requisitos no funcionales

- RNF1: El sistema debe ser eficiente en el uso de recursos.
  - RNF2: Debe ser seguro y evitar fugas de memoria.
  - RNF3: Debe ser extensible para soportar nuevas fuentes de datos o interfaces.
  - RNF4: Debe contar con documentación clara y ejemplos.
- 

### 4. Casos de Uso

- **Desarrolladores:** Analizar el comportamiento de aplicaciones durante pruebas de rendimiento.
  - **SRE / DevOps:** Supervisar servicios y recibir alertas ante límites críticos.
  - **Educación:** Visualizar el uso de recursos para fines didácticos.
  - **CI:** Verificar el consumo de recursos durante compilación o testing.
- 

### 5. Otros Requisitos

- El sistema debe facilitar la integración en flujos de trabajo existentes.
  - Debe permitir la adición de nuevas métricas (uso de red, temperatura, etc.) y nuevas interfaces (web, embebido).
-

## **6. Aprobación**

Este documento será revisado y aprobado por el equipo de desarrollo y las partes interesadas antes de iniciar la implementación.

---