

Especificación de Diseño de Software (EDD)

Sistema: Argos

1. Introducción

1.1 Propósito

Este documento describe el diseño arquitectónico y detallado del sistema Argos, conforme a los requisitos establecidos en la ERS. Está dirigido a desarrolladores, arquitectos de software y personal de mantenimiento.

1.2 Alcance

El EDD cubre la estructura general, los componentes principales, las interfaces, y los mecanismos de interacción del sistema Argos.

1.3 Referencias

- IEEE 1016-2009, IEEE Standard for Information Technology—Systems Design—Software Design Descriptions
 - ERS de Argos (versión actual)
-

2. Descripción General del Diseño

2.1 Perspectiva del producto

Argos se compone de una arquitectura modular, con un núcleo (**argos-core**) y módulos para CLI, logging, exportación de datos y alertas. El sistema está diseñado para ser multiplataforma y extensible.

2.2 Suposiciones y dependencias

- El sistema operativo proporciona acceso a información de procesos.
 - Rust y sus dependencias están correctamente instaladas.
-

3. Arquitectura del Sistema

3.1 Vista de componentes

- **argos-core**: Biblioteca principal que gestiona la recolección de métricas y lógica de negocio.
- **CLI**: Interfaz de línea de comandos para interacción con el usuario.
- **Módulo de Logging**: Encargado de registrar eventos y métricas.
- **Módulo de Exportación**: Permite exportar datos en formatos JSON y CSV.
- **Módulo de Alertas**: Gestiona la definición y notificación de umbrales.
- **FFI Layer**: Permite integración con otros lenguajes (Python, C).

3.2 Diagrama de arquitectura

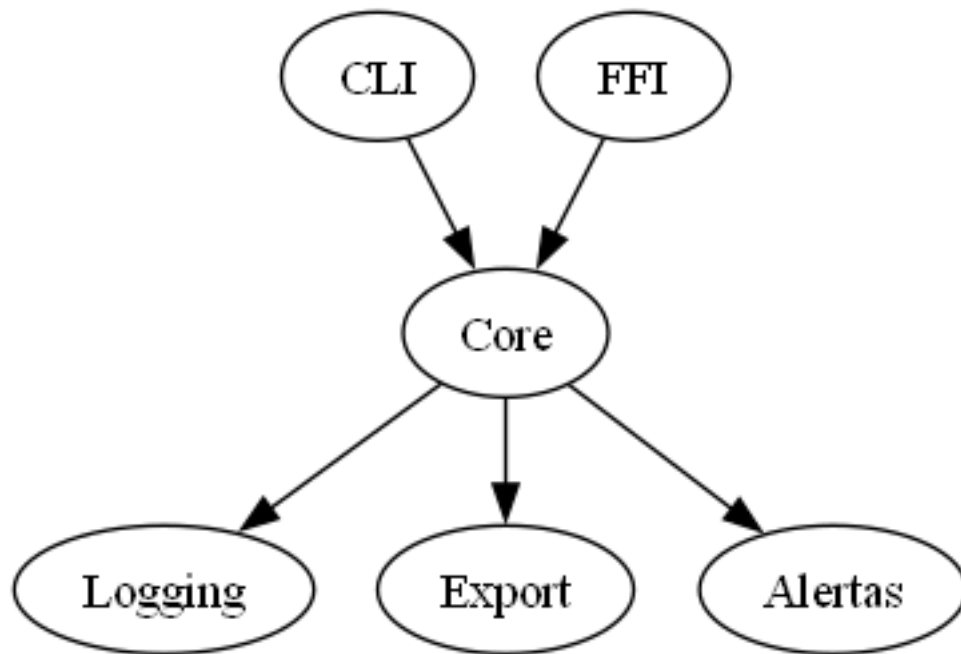


Figure 1: Diagrama de arquitectura de Argos

4. Diseño de Componentes

4.1 argos-core

- **Responsabilidad:** Recolectar métricas de procesos, gestionar intervalos de muestreo, aplicar lógica de umbrales.
- **Interfaces:** API pública para CLI, FFI y módulos externos.
- **Principales estructuras:** `ProcessMonitor`, `MetricsCollector`, `ThresholdManager`.

4.2 CLI

- **Responsabilidad:** Recibir comandos del usuario, mostrar resultados y alertas.
- **Interfaz:** Argumentos de línea de comandos, salida estándar.

4.3 Logging

- **Responsabilidad:** Registrar eventos y métricas.
- **Interfaz:** Funciones para registrar en archivos o consola.

4.4 Exportación

- **Responsabilidad:** Exportar datos en JSON y CSV.
- **Interfaz:** Métodos para serializar y guardar métricas.

4.5 Alertas

- **Responsabilidad:** Definir umbrales y notificar cuando se superan.
- **Interfaz:** Callbacks o eventos para notificación.

4.6 FFI

- **Responsabilidad:** Exponer funciones de `argos-core` a otros lenguajes.
 - **Interfaz:** API compatible con C y Python.
-

5. Interfaces Externas

- **Usuario:** CLI, archivos de configuración.
 - **Sistemas externos:** Integración vía FFI.
-

6. Consideraciones de Diseño

- Modularidad para facilitar la extensión.
 - Uso eficiente de recursos.
 - Seguridad en el manejo de memoria.
 - Portabilidad entre sistemas operativos.
-

7. Glosario

- **CLI:** Interfaz de Línea de Comandos
 - **FFI:** Foreign Function Interface
-

8. Aprobación

Este documento será revisado y aprobado por el equipo de desarrollo antes de la implementación detallada.
