



# Documento Descriptivo: Argos

Herramienta de Monitoreo y Auditoría de Sistemas

---

Ari Castillo

Ingeniería Cibernética Electrónica

 [github.com/Void-CA](https://github.com/Void-CA)

Agosto 2025

Versión 1.0

2025 Ari Castillo - Todos los derechos reservados

## **Síntesis inicial**

Argos es una herramienta multiplataforma de monitoreo y auditoría diseñada para proporcionar visibilidad en tiempo real y análisis histórico del uso de recursos computacionales en sistemas operativos. Desarrollada en Rust para garantizar seguridad, concurrencia y eficiencia, Argos ofrece herramientas avanzadas para administradores de sistemas, ingenieros de DevOps y equipos de seguridad. Su arquitectura modular soporta interfaces CLI y TUI, con planes para integrar una API REST y una interfaz web, facilitando el monitoreo de procesos, detección de anomalías, auditoría histórica y optimización de recursos en entornos críticos.

## Introducción

En el entorno actual de infraestructura crítica, el monitoreo continuo y el análisis detallado de recursos son esenciales para garantizar rendimiento, seguridad y trazabilidad. Argos aborda estas necesidades mediante una plataforma unificada que combina:

- Monitoreo en tiempo real: Visualización de métricas de CPU, memoria, I/O y red por proceso.
- Auditoría histórica: Registro completo para análisis forense y optimización.
- Detección de anomalías: Identificación proactiva de comportamientos inusuales.
- Extensibilidad: Diseño modular compatible con Windows, Linux y macOS.

El núcleo de Argos se apoya en la biblioteca sysinfo, que actúa como el motor principal para la recolección de métricas del sistema y de los procesos. La elección de sysinfo responde a su robustez, eficiencia y, sobre todo, a su capacidad de operar de manera uniforme en múltiples sistemas operativos, permitiendo que Argos ofrezca una experiencia consistente y confiable tanto en entornos Windows, Linux como macOS.

Gracias a sysinfo, Argos puede acceder a información detallada sobre procesos, uso de CPU y memoria, actividad de disco, red y usuarios, garantizando una visión integral y precisa del estado del sistema. Esta base tecnológica facilita la trazabilidad, el análisis histórico y la exportación de datos, posicionando a Argos como una solución integral para entornos que van desde estaciones de trabajo individuales hasta centros de datos distribuidos.

## Objetivos

### Objetivo General

Desarrollar un sistema multiplataforma de supervisión y auditoría que facilite el análisis de eficiencia, trazabilidad y optimización del rendimiento de software, basado en conceptos de sistemas operativos (gestión de procesos, memoria, CPU, sistemas de archivos).

### Objetivos Específicos

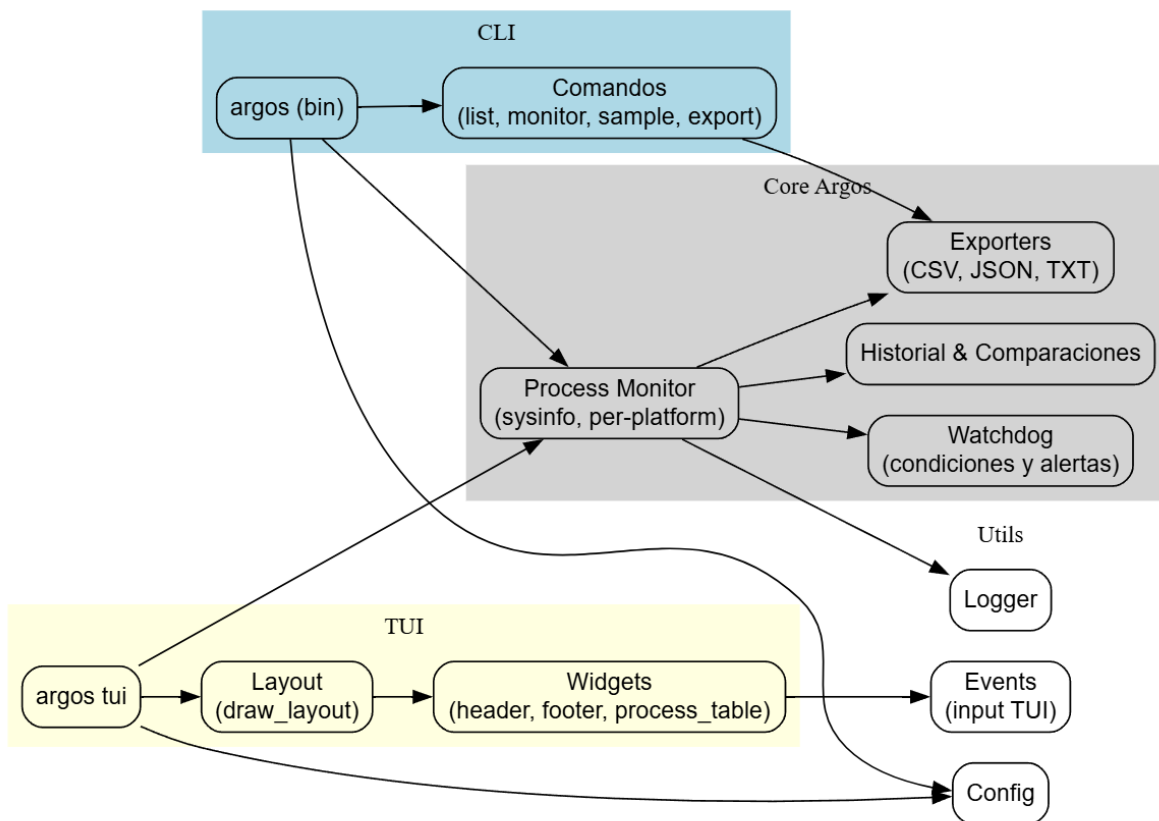
- Monitoreo de procesos: Listar, observar en vivo y muestrear procesos con métricas detalladas (CPU, memoria, estado, usuario).
- Comparación y análisis: Comparar rendimiento entre ejecuciones o procesos para identificar variaciones.
- Exportación de datos: Generar reportes en formatos como CSV, JSON y XML.
- Auditoría histórica: Registrar métricas continuas para análisis longitudinal.
- Extensibilidad: Diseñar una arquitectura modular para futuras integraciones (TUI, GUI, dashboards web).
- Notificaciones proactivas: Implementar alertas basadas en umbrales configurables.

## Arquitectura

La arquitectura de Argos fue concebida bajo el principio de modularidad y reutilización. Cada comando no es un bloque aislado, sino una extensión de un núcleo común. Por ejemplo, monitor constituye la base de la observación de procesos; sobre él se construyen sample (muestreo temporal) y live (seguimiento continuo). De manera análoga, list funciona como el inventario principal de procesos, del cual se derivan family (exploración jerárquica) y compare (contraste entre procesos). Esta interdependencia refuerza la coherencia del diseño y permite ampliar funcionalidades sin duplicar lógica.

Argos se basa en una arquitectura modular que incluye:

- Argos-core: Biblioteca central para recolección de métricas y lógica de negocio.
- CLI: Interfaz de línea de comandos con comandos como monitor, sample, list, history, compare, watchdog, tag y family.
- TUI: Interfaz interactiva basada en ratatui con dashboards y gráficos ASCII.
- Módulo de Exportación: Soporte para JSON, CSV, XML, Parquet e InfluxDB.



## **Tecnologías**

- Backend: Rust (sysinfo, clap).
- Frontend: Rust (ratatui), extensible.

## Requerimientos

### Requerimientos Funcionales

- RF1: Monitoreo en tiempo real de métricas de procesos (CPU%, memoria, estado, puertos).
- RF2: Registro de eventos (creación, terminación, cambios de estado) con consultas históricas.
- RF3: Detección de anomalías con umbrales configurables.
- RF4: Exportación de datos en múltiples formatos y generación de reportes.
- RF5: Control granular de procesos (suspender, reanudar, terminar, priorizar).
- RF6: Interfaces CLI, TUI, y futura API REST/web.

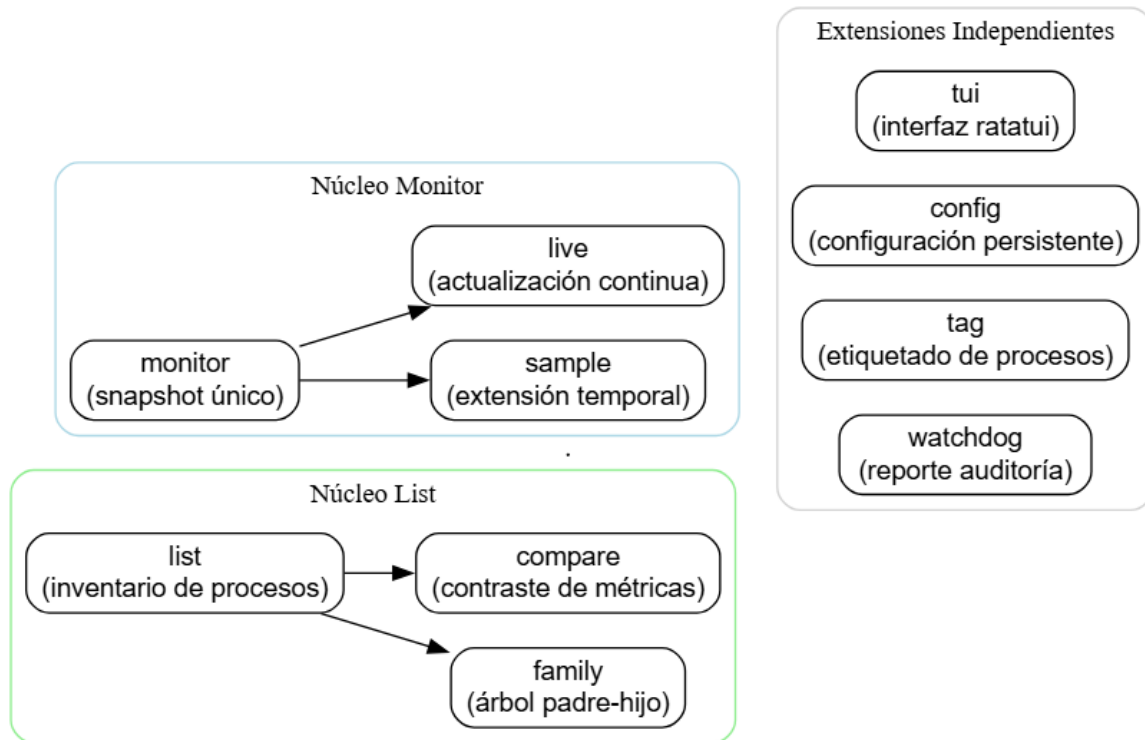
### Requerimientos No Funcionales

- Rendimiento: Overhead < 2% CPU, < 100 MB RAM; latencia < 100 ms.
- Seguridad: Cifrado TLS 1.3, cumplimiento con ISO 27001, GDPR, SOX.
- Escalabilidad: Soporte para 1 a 100,000+ procesos en modo distribuido.
- Compatibilidad: Windows 10+, Linux (kernel 4.0+), macOS 10.14+.

### Requisitos del Sistema

Aspecto	Mínimos	Recomendados
CPU	2 núcleos @ 1.8 GHz	4+ núcleos @ 2.4 GHz
RAM	2 GB	4+ GB
Disco	1 GB	2 GB
SO	Windows 10+, Linux (kernel 4.0+), macOS 10.14+	Igual que mínimos

## Funcionalidades Principales



## Monitoreo en tiempo real

- Comandos: monitor (snapshot), sample (muestreo), list (procesos activos), live (actualización continua).
- Métricas: PID, CPU%, memoria física/virtual, archivos abiertos, conexiones de red.
- Filtros avanzados: Regex, rangos (CPU > 80%), usuario, estado.

## Auditoría histórica

- Registro de eventos: Creación/terminación de procesos, picos de recursos, cambios de permisos.

## Exportación y análisis

- Formatos: CSV, JSON, txt.



## Interfaces

- CLI: Autocompletado, scripting, comandos como `argos monitor --pid 1234 --format csv`.
- TUI: Dashboards interactivos, gráficos ASCII, navegación por teclado.
- Futuras: API REST, interfaz web con D3.js y WebSockets.

```
• argos-demo$ argos -h
Herramienta de auditoría y análisis de recursos en tiempo real

Usage: argos.exe <COMMAND>

Commands:
monitor  Muestra un snapshot del proceso
sample   Realiza un muestreo durante varios segundos
history  Muestra historial de procesos
list     Lista todos los procesos activos
config   Configuración del sistema
live     Muestra información en tiempo real de un proceso
compare  Comparacion entre dos procesos
watchdog Genera logs y reportes de auditoría
tag      Etiquetado de procesos
family   Exploracion de procesos por familia
tui      Punto de entrada de la TUI
help     Print this message or the help of the given subcommand(s)

Options:
-h, --help    Print help
-V, --version Print version
○ argos-demo$
```

## Ejemplos Visuales

### Listado de procesos

Visualización de procesos activos con métricas clave.

Ejemplo: `argos list --top 10 --sort-by memory`.

```
● argos-demo$ argos list -h
Lista todos los procesos activos

Usage: argos.exe list [OPTIONS]

Options:
  -n, --name <NAME>      Filtrar por nombre de proceso
  -u, --user <USER>       Filtrar por usuario
      --sort-by <SORT_BY> Ordenar por (cpu, memory, name, pid) [default: cpu]
  -f, --format <FORMAT>  Formato de salida (text, json, csv) [default: text]
  -o, --output <OUTPUT>  Archivo de salida (opcional)
  -t, --top <TOP>        Limitar número de resultados
  -h, --help              Print help
○ argos-demo$
```

```
● argos-demo$ argos list --top 10 --sort-by memory
PID  Nombre          CPU % RAM MB  Usuario Grupos Estado Inicio  Padre VMEM      Lectura Escritura
-----
19420 msedge.exe    0.00 54.70 void          Run 16:32:28 5984 35.65      5.34 0.03
1876  services.exe     0.00 19.93 -            -    Run 00:00:00 1740 2101358.87 0.00 0.00
3660  svchost.exe      0.00 18.65 -            -    Run 00:00:00 1876 2105548.82 0.00 0.00
20972 svchost.exe      0.00 17.90 -            -    Run 00:00:00 1876 2101361.29 0.00 0.00
3596  svchost.exe      0.00 14.54 -            -    Run 00:00:00 1876 2101347.01 0.00 0.00
10948 svchost.exe      0.00 11.48 -            -    Run 00:00:00 1876 2101337.11 0.00 0.00
9228  svchost.exe      0.00 11.04 void          Run 16:31:22 1876 2.09       0.00 0.00
3100  svchost.exe      0.00 9.87  -            -    Run 00:00:00 1876 2101337.44 0.00 0.00
3900  atiesrxx.exe     0.00 9.56  -            -    Run 00:00:00 1876 4191.89    0.00 0.00
3892  AsusCertService.exe 0.00 8.51  -            -    Run 00:00:00 1876 4183.51    0.00 0.00
○ argos-demo$
```

## Monitoreo de un proceso

Snapshot detallado de un proceso específico.

Ejemplo: `argos monitor --pid 9324`.

```
● argos-demo$ argos monitor -h
Muestra un snapshot del proceso

Usage: argos.exe monitor [OPTIONS] --pid <PID>

Options:
  -p, --pid <PID>      ID del proceso (PID)
  -f, --format <FORMAT> Formato de salida (text, json, csv) [default: text]
  --save               Guardar en base de datos
  -h, --help           Print help
○ argos-demo$
```

```
⊗ argos-demo$ argos monitor --pid 9342
Error interno: Error de proceso no encontrado: [9342]
● argos-demo$ argos monitor --pid 9324
Información del Proceso
Nombre      : opera.exe
PID         : 9324
Estado      : Run
CPU         : 0.00%
RAM         : 114.14 MB
Inicio      : 1756917251
PID Padre   : 16300
○ argos-demo$
```

## Muestreo

Análisis temporal de un proceso durante varios segundos.

Ejemplo: `argos sample --pid 18044`.

```
● argos-demo$ argos sample -h
Realiza un muestreo durante varios segundos

Usage: argos.exe sample [OPTIONS] --pid <PID>

Options:
  -p, --pid <PID>          ID del proceso (PID)
  --iterations <ITERATIONS> Número de iteraciones [default: 10]
  -i, --interval-ms <INTERVAL_MS> Intervalo entre muestras (milisegundos) [default: 200]
  -f, --format <FORMAT>    Formato de salida (text, json, csv) [default: text]
  -o, --output <OUTPUT>    Archivo de salida (opcional)
  -h, --help                Print help
○ argos-demo$
```

```
● argos-demo$ argos sample --pid 18044
Timestamp  PID Nombre  CPU % RAM MB Estado
-----
16:31:45  18044 Code.exe  0.00 172.23 Run
16:31:45  18044 Code.exe  0.00 172.21 Run
16:31:45  18044 Code.exe  0.00 172.21 Run
16:31:45  18044 Code.exe  0.00 172.21 Run
16:31:45  18044 Code.exe  0.00 172.21 Run
16:31:45  18044 Code.exe  0.00 172.21 Run
16:31:45  18044 Code.exe  0.00 172.21 Run
16:31:45  18044 Code.exe  0.00 172.21 Run
16:31:45  18044 Code.exe  0.00 172.21 Run
16:31:45  18044 Code.exe  0.00 172.21 Run
16:31:45  18044 Code.exe  0.00 172.21 Run

Estadísticas:
CPU % -> Promedio: 0.00, Min: 0.00, Max: 0.00, Desv.Est: 0.00
RAM MB -> Promedio: 172.21, Min: 172.21, Max: 172.23, Desv.Est: 0.01
○ argos-demo$
```

## Interfaz TUI

## Navegación de procesos.

La interfaz TUI de Argos permite explorar de manera interactiva la lista de procesos activos del sistema, mostrando información relevante y actualizada en tiempo real. Cada proceso se presenta en una tabla con los siguientes campos:

PID: Identificador único del proceso en el sistema. **Nombre:** Nombre del ejecutable o comando asociado al proceso. **CPU %:** Porcentaje de uso de CPU en el instante actual. **Memoria:** Cantidad de memoria física utilizada (MB). **Usuario:** Usuario propietario del proceso. **Estado:** Estado actual del proceso (ejecutando, suspendido, zombie, etc.). **Lectura Disco:** Volumen de datos leídos por el proceso (MB/s). **Escritura Disco:** Volumen de datos escritos por el proceso (MB/s). **Inicio:** Fecha y hora de inicio del proceso. **PID Padre:** Identificador del proceso padre (si existe). **Memoria Virtual:** Cantidad de memoria virtual asignada (MB).

La navegación se realiza mediante el teclado, permitiendo desplazarse por la lista, buscar procesos por nombre o usuario, y aplicar filtros dinámicos. La tabla se actualiza automáticamente, reflejando cambios en el estado, recursos y actividad de cada proceso. Esta vista facilita la identificación de procesos críticos, el monitoreo de recursos y la toma de decisiones sobre la gestión y optimización del sistema.

Argos TUI - 287 procesos | Actualizando cada 2s

Procesos

PID	Nombre	CPU%	Memoria	Usuario	Estado	Lectura Disco	Escritura Disco	Inicio	PID Padre	Memoria Virtual
>> 5124	spoolsv.exe	0.00	21.15 MB	-	Run	0.00	0.00	00:00:00	1876	2101360.17 MB
3976	opera.exe	0.00	227.63 MB	void	Run	10.07	1.93	16:34:10	16300	101.28 MB
3900	atiesxxx.exe	0.00	9.56 MB	-	Run	0.00	0.00	00:00:00	1876	4191.89 MB
20516	svchost.exe	0.00	8.85 MB	-	Run	0.00	0.00	00:00:00	1876	2101331.79 MB
5756	svchost.exe	0.00	21.87 MB	-	Run	0.00	0.00	00:00:00	1876	2101373.62 MB
16120	ArmouryCrat.exe	0.00	5.21 MB	void	Run	0.01	0.00	16:31:26	1012	45.20 MB
5984	msedge.exe	0.00	127.05 MB	void	Run	16.41	9.69	16:32:28	6968	46.18 MB
4668	svchost.exe	0.00	28.06 MB	-	Run	0.00	0.00	00:00:00	1876	2101388.71 MB
2792	svchost.exe	0.00	10.60 MB	-	Run	0.00	0.00	00:00:00	1876	2101343.75 MB
22824	conhost.exe	0.00	7.19 MB	void	Run	0.00	0.00	16:47:58	3804	1.17 MB
4936	svchost.exe	0.00	12.72 MB	-	Run	0.00	0.00	00:00:00	1876	2101344.07 MB
9564	ASMonitorControl.exe	0.00	1.87 MB	void	Run	0.00	0.00	16:31:22	2736	1.62 MB
13536	opera.exe	0.00	24.14 MB	void	Run	0.13	0.13	16:34:11	16300	8.21 MB
21676	opera.exe	0.00	140.34 MB	void	Run	1.85	1.72	16:34:24	16300	46.43 MB
7776	postgres.exe	0.00	23.02 MB	-	Run	0.00	0.00	00:00:00	5832	4408.88 MB
1924	svchost.exe	0.00	11.15 MB	void	Run	0.00	0.00	16:31:22	1876	2.12 MB
2336	svchost.exe	0.00	11.21 MB	-	Run	0.00	0.00	00:00:00	1876	2101333.27 MB
5616	ArmouryCrat.Service	0.00	59.41 MB	-	Run	0.00	0.00	00:00:00	1876	2101996.28 MB
13444	msedgeview2.exe	0.00	9.09 MB	void	Run	0.00	0.00	16:31:25	9980	2.25 MB
9584	svchost.exe	0.00	20.74 MB	-	Run	0.00	0.00	00:00:00	1876	2101370.85 MB
15920	svchost.exe	0.00	20.11 MB	-	Run	0.00	0.00	00:00:00	1876	2101368.37 MB
11316	opera.exe	0.00	134.86 MB	void	Run	1.15	1.08	16:34:16	16300	47.35 MB
2988	svchost.exe	0.00	7.98 MB	-	Run	0.00	0.00	00:00:00	1876	2101336.30 MB
19176	backgroundTaskHost.exe	0.00	26.64 MB	void	Run	0.00	0.00	17:06:37	1012	7.56 MB
9980	msedgeview2.exe	0.00	114.42 MB	void	Run	17.41	6.14	16:31:25	11760	35.98 MB
20600	opera.exe	0.00	130.09 MB	void	Run	0.56	1.60	16:34:16	16300	41.35 MB
11568	RuntimeBroker.exe	0.00	42.54 MB	void	Run	0.14	0.00	16:31:23	1012	6.54 MB
4720	Code.exe	5.95	357.20 MB	void	Run	161.02	417.83	16:47:51	18044	304.83 MB
8484	Code.exe	0.00	157.04 MB	void	Run	0.29	0.26	17:06:00	18044	64.35 MB
7828	postgres.exe	0.00	11.25 MB	-	Run	0.00	0.00	00:00:00	7776	4368.54 MB
5288	svchost.exe	0.00	9.36 MB	-	Run	0.00	0.00	00:00:00	1876	2101329.57 MB
4664	Code.exe	0.00	124.68 MB	void	Run	0.20	0.15	17:06:00	18044	34.98 MB
21324	opera.exe	0.00	57.34 MB	void	Run	0.08	0.05	16:34:19	16300	36.07 MB
7888	conhost.exe	0.00	9.18 MB	-	Run	0.00	0.00	00:00:00	7776	2101332.48 MB
1984	LsaIso.exe	0.00	4.22 MB	-	Run	0.00	0.00	00:00:00	1740	4146.61 MB

q:Salir | r:Actualizar | f:Buscar | f:Filtrar | c:Limpiar | ↕:Navegar | Última: 3s

## Detalles de un proceso

La vista de detalles en la TUI de Argos permite analizar en profundidad el comportamiento de cualquier proceso seleccionado. Para cada proceso, se muestran métricas históricas y estadísticas avanzadas, calculadas en tiempo real y presentadas de forma clara y visual.

Métricas presentadas:

- CPU:
  - Historial de uso porcentual de CPU.
  - Estadísticas: media, mediana, desviación estándar, mínimo, máximo, percentil 95 y sesgo.
  - Algoritmo de tendencia: se calcula la pendiente de la regresión lineal sobre el historial, permitiendo identificar si el uso de CPU está aumentando, disminuyendo o es estable.
  - Predicción: se estima el próximo valor esperado de uso de CPU.
- Memoria:
  - Historial de consumo de memoria (MB).
  - Estadísticas: media, mediana, desviación estándar, mínimo, máximo, percentil 95 y sesgo.
  - Tendencia: análisis de la evolución del consumo de memoria, útil para detectar fugas o picos inesperados.
  - Predicción: estimación del próximo valor de memoria.
- Disco:
  - Historial de operaciones de lectura y escritura (MB/s).
  - Estadísticas separadas para lectura y escritura: media, mediana, desviación estándar, mínimo, máximo, percentil 95 y sesgo.
  - Tendencia: permite identificar patrones de acceso a disco, como cargas intensivas o periodos de inactividad.
  - Predicción: próximo valor esperado de actividad de disco.

Análisis estadístico:

Para cada métrica, Argos calcula y muestra: - Media y mediana: valores centrales del historial. - Desviación estándar: variabilidad de la métrica. - Mínimo y máximo: extremos observados. - Percentil 95: valor por debajo del cual se encuentra el 95% de las muestras. - Sesgo (skewness): indica si la distribución de los valores está inclinada hacia valores altos o bajos. - Tendencia: pendiente de la regresión lineal, útil para anticipar comportamientos futuros. - Predicción: estimación del próximo valor, basada en el modelo de tendencia.

Utilidad práctica: - El usuario puede identificar rápidamente procesos problemáticos, tendencias de consumo anómalo, picos de actividad o posibles fugas de recursos. - El análisis estadístico

detallado permite tomar decisiones informadas sobre la gestión de procesos y recursos, facilitando la optimización y la prevención de incidentes.

Detalles del Proceso - PID: 17516 | Nombre: Code.exe

Estadísticas de CPU

CPU: 3.0%  
Media: 2.0%  
Mediana: 3.0%  
o: 2.1%  
Min: 0.0%  
Máx: 6.0%  
Percentil 95: 6.0%

Estadísticas de Memoria

Memoria: 186.1 MB  
Media: 193.0 MB  
Mediana: 194.4 MB  
o: 3.2 MB  
Min: 186.1 MB  
Máx: 194.5 MB  
Percentil 95: 194.5 MB

Estadísticas de Disco

Lectura: 1247.5 MB/s  
Escritura: 23.9 MB/s  
o Lectura: 0.2  
o Escritura: 0.0  
Tendencia L: 1.795 MB/s/min  
Tendencia E: 0.219 MB/s/min  
P95 L: 1247.5

Análisis Estadístico Detallado

Estab. CPU: -6.4%  
Estab. Mem: 98.4%  
Coef. Var. CPU: 1.064  
Coef. Var. Mem: 0.016  
Sesgo CPU: 0.560  
Sesgo Mem: -1.720  
Coef. Var. L: 0.000

Esc/q:Volver | r:Actualizar estadísticas

## **Roadmap de Implementación**

### **Completados**

- Núcleo de monitoreo: Recolecta métricas de todos los procesos activos del sistema en tiempo real, utilizando técnicas eficientes y seguras para minimizar el impacto en el rendimiento.
- CLI y TUI: Proveen interfaces interactivas y de línea de comandos para consultar, filtrar y visualizar la información de procesos, recursos y eventos. La TUI incluye dashboards, métricas descriptivas, algoritmos de tendencias y navegación por teclado.
- Módulo de exportación: Permite generar archivos en formatos estándar y enviar datos a sistemas externos.

### **En desarrollo futuro**

- Módulo de auditoría automatizada: Registra y almacena todos los eventos relevantes en una base de datos local, permitiendo consultas históricas y generación de reportes.
- Gestión de procesos: El usuario puede suspender, reanudar, terminar o cambiar la prioridad de cualquier proceso desde la interfaz.



## Puntos de Mejora

- Persistencia: Implementar SQLite para historial completo.
- CLI avanzada: Completar subcomandos watchdog, history, tag.
- TUI: Agregar filtros dinámicos y gráficos en tiempo real.
- Alertas: Configuración avanzada de umbrales y notificaciones.
- Integración: Desarrollar API REST y paneles web.

## Consideraciones Técnicas

Durante el desarrollo, se aplicaron conceptos de sincronización, planificación de procesos y administración de recursos estudiados en la asignatura de Sistemas Operativos II. Argos no solo funciona como herramienta práctica, sino también como un laboratorio de experimentación, donde se tradujeron fundamentos teóricos en software real, eficiente y portable.

## Rendimiento objetivo

- Overhead:  $< 2\%$  CPU,  $< 100$  MB RAM.
- Latencia:  $< 100$  ms para consultas simples.
- Throughput:  $> 10,000$  eventos/segundo.

## Despliegue

Argos se distribuye como un binario único para cada sistema operativo soportado (Windows, Linux, macOS). La instalación es sencilla y no requiere dependencias externas: basta con descargar el ejecutable y ejecutarlo. La configuración se realiza mediante archivos YAML o directamente desde la CLI, permitiendo adaptar el comportamiento del sistema a las necesidades del entorno.

## Costos y Requerimientos de Infraestructura

- Hardware: Compatible con equipos de gama baja (2 núcleos, 2 GB RAM) y alta.
- Software: Licencia de código abierto, costo de desarrollo mínimo.
- Mantenimiento: Revisión trimestral, actualizaciones automáticas.

## Conclusión

Argos es una solución consolidada para la supervisión y auditoría de sistemas, que ya integra monitoreo en tiempo real, análisis histórico y detección de anomalías en una plataforma modular y eficiente. Gracias a su núcleo desarrollado en Rust, el sistema garantiza rendimiento óptimo, seguridad robusta y portabilidad multiplataforma, permitiendo su uso tanto en estaciones de trabajo como en infraestructuras empresariales.

El usuario dispone de potentes herramientas para el análisis detallado de procesos, con métricas avanzadas y trazabilidad completa de eventos. La exportación flexible en formatos como CSV, JSON y TXT facilita la integración con sistemas externos, el procesamiento en hojas de cálculo y la generación de reportes personalizados. La auditoría histórica y la correlación de datos permiten reconstruir la actividad de cualquier proceso, identificar patrones de uso y anticipar incidentes, aportando valor tanto en la operación diaria como en el análisis forense.

La arquitectura modular de Argos permite ampliar el alcance del monitoreo más allá de los procesos, incorporando información sobre usuarios, red, almacenamiento y otros recursos del sistema, aprovechando las capacidades de sysinfo y otras fuentes de datos. Además, el diseño del sistema facilita la integración con nuevas interfaces, como paneles web, aplicaciones GUI y APIs REST, abriendo la puerta a escenarios de visualización avanzada, automatización y orquestación.

En síntesis, Argos no solo ofrece visibilidad y control sobre los recursos del sistema, sino que proporciona una base sólida para el análisis predictivo, la optimización de recursos y la evolución hacia entornos de monitoreo e integración cada vez más completos y escalables.

## Referencias

- [1] IEEE Std 1016-2009, IEEE Standard for Information Technology—Systems Design—Software Design Descriptions, IEEE, Piscataway, NJ, USA, 2009.
- [2] IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE, Piscataway, NJ, USA, 1998.
- [3] G. Gomez, “sysinfo: Cross-platform library to fetch system information,” docs.rs, 2023. [Online]. Available: <https://docs.rs/sysinfo>.
- [4] Ratatui Development Team, “ratatui: Rust library for building rich terminal user interfaces,” docs.rs, 2023. [Online]. Available: <https://docs.rs/ratatui>.
- [5] Clap Development Team, “clap: Command Line Argument Parser for Rust,” docs.rs, 2023. [Online]. Available: <https://docs.rs/clap>.
- [6] A. Castillo, “Documentación ERS de Argos,” Internal Project Documentation, Universidad Tecnológica La Salle, Leon, Nicaragua, 2025.