

# Documentación Tarea Daemon

Andres Uriza-Lazo Ingenieria en computadores

Instituto Tecnológico de Costa Rica

Email: urizandres@estudiantec.cr

Daniel Duarte-Cordero Ingenieria en computadores

Instituto Tecnológico de Costa Rica

Email: daduarte@estudiantec.cr

**Abstract**—This project presents the development of a Linux daemon named ImageServer, responsible for receiving images via TCP sockets, processing them using histogram equalization techniques, and classifying them based on their dominant color. The solution was implemented in the C programming language, incorporating concepts of concurrent programming, client-server communication, and background service management through systemd. Additionally, a client application running inside a container was designed to allow users to select images for analysis. The project reinforces key technical skills and promotes best practices in collaborative work, planning, and documentation under a continuous learning approach.

**Palabras clave**—Linux Daemon, Client-Server Architecture, TCP Sockets, Histogram Equalization, Image Classification, systemd, C Programming, Docker, Concurrent Programming, Configuration Files

## I. APRENDIZAJE CONTINUO

*A. ¿Cuáles son las necesidades actuales de aprendizaje para enfrentar el proyecto?*

El desarrollo del proyecto evidenció la necesidad de fortalecer varios conocimientos clave. En primer lugar, se identificó la importancia de mejorar el dominio del lenguaje C, particularmente en aspectos de programación modular, gestión de memoria y manejo de procesos concurrentes. También fue necesario comprender el funcionamiento de los demonios en Linux, lo que implica aprender cómo se crean procesos en segundo plano y cómo se integran al arranque del sistema operativo.

Otro punto de aprendizaje consistió en profundizar en los fundamentos del protocolo TCP y en la lógica de comunicación entre cliente y servidor, de manera que la implementación resultara robusta y confiable. Asimismo, se reconoció la necesidad de adquirir habilidades para trabajar con entornos de virtualización y contenedores, ya que estos permiten separar la ejecución de los componentes del sistema. Finalmente, se requirió afianzar el uso de herramientas de automatización y de archivos de configuración, con el fin de lograr un flujo de trabajo más ordenado y reproducible.

*B. ¿Cuáles son las tecnologías que se pueden utilizar para el desarrollo?*

El proyecto integra diversas tecnologías que garantizan su correcto funcionamiento. La base es el lenguaje C junto con librerías estándar para la comunicación mediante sockets, el

protocolo TCP/IP y la creación de hilos, como `sys/socket.h`, `netinet/in.h` y `pthread.h`. Para el despliegue se utiliza Docker, que permite ejecutar el cliente en un entorno aislado y reproducible. En la gestión del daemon se consideran herramientas nativas de Linux como systemd o SysVinit, que facilitan iniciar, detener o reiniciar el servicio.

Además, se emplean makefiles para automatizar la compilación y archivos de configuración para definir parámetros como rutas, puertos y directorios de salida. En caso de ampliar las funciones, pueden integrarse librerías de imágenes como libjpeg o libpng. Con ello, se logra una base sólida y flexible para cumplir los objetivos del sistema.

*C. ¿Cuáles acciones se implementó para el desarrollo del proyecto (organización de tiempo, búsqueda de información, repaso de contenidos, entre otros)?*

Durante el desarrollo del proyecto se implementaron diversas acciones orientadas a la organización y al fortalecimiento del conocimiento técnico. En primer lugar, se recurrió a la lectura de documentación oficial del lenguaje C y de sus librerías estándar, con el fin de comprender de manera más profunda el uso de sockets, hilos y manejo de procesos. Como complemento, se realizó un repaso de conceptos previamente vistos en el curso, apoyándose en herramientas de inteligencia artificial para resolver dudas puntuales y recuperar contenidos que habían sido olvidados.

En cuanto a la organización del trabajo, se adoptó una dinámica inspirada en la metodología SCRUM, dividiendo el avance en épicas y tareas específicas, gestionadas en una tabla con los estados de pendiente, en progreso y finalizado. Finalmente, como parte de la investigación aplicada, se revisaron repositorios similares en GitHub, lo que permitió identificar buenas prácticas de implementación y comparar distintas estrategias utilizadas por otros desarrolladores en proyectos afines.

*D. Evalúe de forma crítica la eficiencia de las acciones implementadas en el contexto tecnológico.*

La lectura de documentación y el repaso de conceptos apoyado en inteligencia artificial resultaron acciones muy efectivas, ya que permitieron consolidar un conocimiento robusto previo al desarrollo. Estas herramientas facilitaron la comprensión de aspectos técnicos esenciales y brindaron seguridad al momento

de implementar el código. Asimismo, la revisión de repositorios en GitHub fue de utilidad para observar buenas prácticas y enfoques distintos, aunque requirió un esfuerzo adicional para adaptarlos a las necesidades específicas del proyecto.

Por otro lado, la división inicial del trabajo no fue la más eficiente. Al no asignar un nivel de dificultad equilibrado a las tareas, se generó una distribución desigual de la carga, lo que obligó a redistribuir responsabilidades en etapas posteriores. Esta situación evidenció la importancia de una mejor planificación al inicio del proyecto. En conjunto, las acciones implementadas permitieron alcanzar los objetivos, aunque se identificaron áreas de mejora en la gestión del trabajo colaborativo.

## II. TRABAJO INDIVIDUAL Y EN EQUIPO

### *A. Indicar cuales acciones promueven la colaboración entre los miembros del equipo durante el desarrollo del proyecto.*

En la planificación se definieron roles claros para cada integrante, lo que permitió un inicio ordenado del proyecto. Durante la ejecución se utilizó una tabla tipo Kanban y un repositorio en GitHub para documentar avances, además de aplicar redistribución de tareas cuando se detectaron cargas desiguales. Finalmente, en la etapa de evaluación se realizó una revisión conjunta del código antes de la entrega, garantizando un proceso equitativo e inclusivo en todas las fases.

### *B. Indicar la planificación del trabajo mediante la identificación de roles, metas y reglas.*

En la planificación se definieron roles específicos para cada integrante. El miembro 1 asumió la responsabilidad del cliente, la programación de sockets, la correcta recepción de archivos en el servidor y la documentación final. El miembro 2 se encargó del desarrollo del histograma de ecualización, la clasificación de imágenes por color y la implementación del daemon en Linux.

Como metas se estableció asegurar la comunicación cliente-servidor en una fase temprana, completar el procesamiento de imágenes antes de la fecha límite y mantener la documentación actualizada durante el desarrollo. Las reglas acordadas incluyeron el uso de un repositorio en GitHub para registrar avances, definir WhatsApp como canal oficial de comunicación, y establecer que, en caso de complicaciones o bloqueos, se informaría de inmediato al otro integrante para abordar el problema de forma conjunta.

### *C. Indicar cómo se ejecutan las estrategias planificadas para el logro de los objetivos.*

Las estrategias definidas en la etapa de planificación se ejecutaron de forma constante a lo largo del proyecto. Cada integrante mantuvo su rol asignado, avanzando en paralelo en las partes que le correspondían, lo que permitió desarrollar cliente y servidor de manera coordinada. Se utilizaron entregas parciales para validar avances, especialmente en la conexión cliente-servidor, evitando problemas de integración al final.

El uso activo de GitHub facilitó la gestión de versiones y permitió que ambos miembros estuvieran al tanto del progreso

general. La comunicación por WhatsApp fue fundamental para resolver dudas rápidamente y coordinar la secuencia de tareas. Estas acciones permitieron cumplir con los objetivos establecidos y responder con flexibilidad a los retos que surgieron durante el desarrollo.

### *D. Indicar la evaluación para la el desempeño del trabajo individual y en equipo*

El desempeño fue evaluado mediante una combinación de autoevaluación, retroalimentación cruzada y revisión de cumplimiento con respecto a los objetivos definidos. Desde el inicio, se acordó una Definition of Done (DoD) para cada tarea, que incluía criterios como funcionalidad completa, pruebas básicas exitosas y claridad en el código. Esta definición permitió evaluar objetivamente si cada entregable individual cumplía con lo esperado antes de marcarlo como finalizado.

A nivel grupal, la evaluación se centró en el cumplimiento de metas parciales, la correcta integración de los módulos y la efectividad en la comunicación. También se consideraron factores como la adaptación ante desequilibrios en la carga de trabajo y la disposición a colaborar. Esta estructura permitió realizar una evaluación más justa y alineada con el desempeño real de cada miembro y del equipo como conjunto.

### *E. Indicar la evaluación para las estrategias utilizadas de equidad e inclusión.*

Las estrategias de equidad e inclusión se evaluaron mediante una revisión conjunta del proceso de trabajo, analizando si la distribución de tareas había sido justa y si ambos integrantes tuvieron oportunidad de aportar en partes clave del proyecto.

También se consideró la participación activa de ambos miembros en la toma de decisiones, la disposición para escuchar propuestas y el respeto mutuo en cuanto a ritmos de trabajo y estilos de comunicación. La evaluación mostró que las estrategias aplicadas fueron efectivas en mantener un ambiente de trabajo colaborativo y respetuoso, aunque se identificó como mejora futura realizar una planificación más precisa para evitar ajustes durante la ejecución.

### *F. Indicar la evaluación para las acciones de colaboración entre los miembros del equipo*

La evaluación de las acciones de colaboración se realizó considerando su efectividad para alcanzar los objetivos del proyecto. Se valoró la sincronización de entregas parciales, ya que permitió que el desarrollo del cliente y del servidor avanzara en paralelo sin generar bloqueos entre módulos. Esta estrategia fue eficaz para evitar cuellos de botella y facilitó una integración técnica fluida.

También se evaluó el uso de WhatsApp como canal de comunicación constante, lo cual se reflejó en una rápida resolución de dudas y en la capacidad de responder de forma oportuna ante imprevistos. El intercambio de código para pruebas cruzadas permitió detectar errores de manera anticipada, y se consideró una práctica colaborativa útil y replicable. En general, la evaluación concluyó que las acciones implementadas fomentaron una colaboración efectiva y tuvieron un impacto directo en la calidad y estabilidad del sistema final.