

Hacia la Evolución de la GUI Desktop: IA Centrada en el Humano como Paradigma para Repensar la Interacción con Sistemas Operativos

José Ramón Aragón Toledo¹, Ricardo Ruiz Rodríguez¹

¹Universidad Tecnológica de la Mixteca (UTM)
Huajuapán de León, Oaxaca – México

aatr010423@gmail.com, rruiz@mixteco.utm.mx

Abstract. *Innovation in desktop operating system interfaces has stagnated over the past two decades. Despite exponential growth in computational power, tools such as file explorers, terminals, and configuration managers have scarcely evolved since the 1990s. This plateau particularly affects GNU/Linux, whose power remains inaccessible to non-expert users due to command-line interaction models from the 1970s. This paper analyzes desktop interaction stagnation through the lens of Human-Centered AI (HCAI) and presents BOTAS, a voice-based assistant enabling Spanish-speaking users to interact with GNU/Linux via natural language. Evaluation across 200 tasks shows 89 % command-generation accuracy and 100 % detection of destructive operations.*

Resumen. *La innovación en las interfaces de sistemas operativos de escritorio ha experimentado un estancamiento en las últimas dos décadas. A pesar del crecimiento exponencial en capacidad computacional, herramientas como exploradores de archivos, terminales y gestores de configuración apenas han evolucionado desde los años noventa. Esta meseta afecta especialmente a GNU/Linux, cuyo potencial permanece inaccesible para usuarios no expertos debido a modelos de interacción de línea de comandos de los años setenta. Este artículo analiza el estancamiento de la interacción con el escritorio a través del marco de la Inteligencia Artificial Centrada en el Humano (HCAI) y presenta BOTAS, un asistente de voz que permite a usuarios hispanohablantes interactuar con GNU/Linux mediante lenguaje natural. La evaluación sobre 200 tareas muestra un 89 % de precisión en la generación de comandos y un 100 % de detección de operaciones destructivas.*

1. Introducción

Operamos con procesadores cien millones de veces más rápidos que los de la década de 1980, y sin embargo la interacción fundamental con un sistema operativo de escritorio apenas ha cambiado en veinte años [Jenson 2024]. Scott Jenson—veterano diseñador de experiencia de usuario que participó en el desarrollo del Finder original de Macintosh en Apple y posteriormente condujo investigación sobre interacción en Google—sostiene que “este inmovilismo no es un límite técnico sino una claudicación del diseño” [Jenson 2024]. La industria ha transitado de la transformación creativa que definió los albores de la computación—donde conceptos de Xerox PARC fueron radicalmente reimaginados—a una imitación superficial entre plataformas.

GNU/Linux sustenta aproximadamente el 90 % de la infraestructura en la nube y el 100 % de las 500 supercomputadoras más potentes del mundo [W3Techs 2023, Top500.org 2023]; no obstante, su interfaz de línea de comandos perpetúa un paradigma de la década de 1970 [Raymond 2003] que exige memorizar sintaxis crípticas sin retroalimentación sobre el riesgo de las operaciones. Mientras tanto, los asistentes de voz comerciales gestionan consultas triviales pero ninguno aborda la interacción profunda con el sistema operativo [Luger and Sellen 2016], y herramientas como ChatGPT sugieren comandos sin ejecutarlos ni validar su seguridad. Este trabajo sostiene que la Inteligencia Artificial Centrada en el Humano (HCAI) [Shneiderman 2020, Shneiderman 2022] ofrece un marco para superar este estancamiento, y presenta BOTAS, un asistente de voz que materializa los principios HCAI para la interacción con GNU/Linux en español.

2. Marco Conceptual y Trabajo Relacionado

2.1. Estancamiento de las interfaces de escritorio

La interacción humano-computadora ha atravesado períodos de innovación genuina—la metáfora de escritorio, la Web, la revolución táctil—que redefinieron los modelos mentales de los usuarios [Norman 1988]. Sin embargo, en el escritorio las últimas dos décadas constituyen una meseta funcional [Jenson 2024]. Estudios recientes confirman que las herramientas integradas en los SO modernos —exploradores de archivos, terminales, gestores de configuración—no han experimentado cambios paradigmáticos desde mediados de los 2000 [Grudin 2017]. Para GNU/Linux, la CLI constituye la barrera principal hacia una plataforma que domina la infraestructura profesional [W3Techs 2023].

2.2. Inteligencia Artificial Centrada en el Humano

Shneiderman [Shneiderman 2020, Shneiderman 2022] formalizó HCAI como un paradigma donde la IA *amplifica* capacidades humanas en lugar de reemplazarlas, articulado sobre cuatro principios: **transparencia** (el usuario comprende las acciones del sistema), **control humano** (supervisión y rechazo de propuestas), **seguridad** (prevención de acciones dañinas, crítico donde un error puede causar pérdida irreversible de datos) y **empoderamiento** (el usuario construye autonomía progresiva). Amershi et al. [Amershi et al. 2019] complementan con 18 directrices prácticas. HCAI provee un cimiento coherente para repensar la interacción con el SO: un asistente fundado en estos principios puede hacer accesible GNU/Linux asegurando que el usuario aprenda en lugar de depender.

2.3. Asistentes de voz y la brecha de interacción con el SO

Los avances en reconocimiento de voz—particularmente Whisper [Radford et al. 2022]—y en modelos de lenguaje de gran escala [Brown et al. 2020, OpenAI 2023] han alcanzado madurez práctica multilingüe. NL2Bash [Lin et al. 2018] exploró la traducción de lenguaje natural a Bash (36 % de precisión exacta) pero sin interacción en tiempo real ni validación de seguridad. Ningún sistema existente integra reconocimiento de voz, ejecución de comandos, clasificación de seguridad y retroalimentación educativa dentro de un flujo orientado por HCAI. Esta brecha de integración define la oportunidad que aborda este trabajo.

3. Propuesta: BOTAS con Enfoque HCAI

Los principios HCAI se materializan sobre GNU/Linux porque es el entorno donde esta intervención genera mayor impacto: un sistema extraordinariamente potente cuya principal barrera de adopción es la interacción, no la funcionalidad [Warschauer 2004]. Su carácter abierto permite integración profunda, y su gratuidad elimina barreras económicas—factor decisivo para contextos educativos en América Latina. BOTAS (*Bot de Operaciones y Tareas Automatizadas del Sistema*) se originó durante un programa de servicio social en una universidad pública mexicana, explorando si el lenguaje natural podía servir como puente entre usuarios novatos y la CLI de GNU/Linux. Desde un MVP dependiente de la nube, se incorporaron iterativamente: detección de palabra de activación fuera de línea (Vosk), reconocimiento local de intenciones, detección automática de distribución en seis familias Linux y registro estructurado.

3.1. Capacidades actuales

BOTAS implementa un pipeline modular cuyo flujo se ilustra en la Figura 1: (1) captura de entrada mediante GUI o palabra de activación; (2) detección local de intenciones por expresiones regulares; (3) análisis por GPT-4o para solicitudes complejas; (4) validación de seguridad determinista en tres niveles (*seguro, advertencia, peligroso*); y (5) ejecución con retroalimentación educativa y TTS. Cada operación muestra el comando generado antes de ejecutarlo (transparencia), explica su función (empoderamiento) y exige confirmación para operaciones riesgosas (control humano). Limitaciones actuales: dependencia de servicios en la nube (Whisper, GPT-4o), dificultad con peticiones ambiguas y ausencia de diálogos multiturnos; se desarrolla operación local con Vosk y modelos ligeros para entornos institucionales.

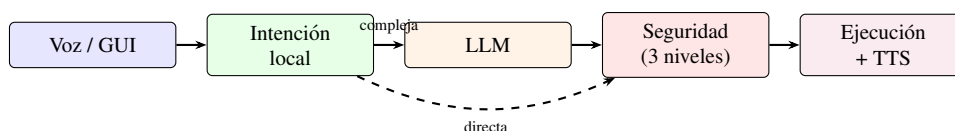


Figura 1. Pipeline de procesamiento de BOTAS. Las peticiones rutinarias (29%) se resuelven localmente sin LLM (línea punteada).

4. Evaluación Preliminar

4.1. Pruebas funcionales

BOTAS fue evaluado con un conjunto de 200 tareas de administración de GNU/Linux en cinco categorías: gestión de archivos (45), información del sistema (40), control de procesos (35), redes (30) y permisos (25). Las tareas fueron seleccionadas a partir de operaciones frecuentes documentadas en manuales de administración de sistemas y foros de soporte, cubriendo tres niveles de riesgo: seguro (169), advertencia (18) y peligroso (13). El conjunto fue compilado y verificado manualmente por el desarrollador principal—restricción metodológica que puede introducir sesgo de confirmación. La Tabla 1 resume los resultados.

Modos de error: fraseo ambiguo (6%), errores de banderas en pipes múltiples (3%), indisponibilidad específica de distribución (2%). La capa de seguridad determinista detectó el 100% de las 43 operaciones peligrosas (eliminación recursiva, escalamiento

Tabla 1. Precisión en la generación de comandos por categoría

Categoría	Tareas	Precisión
Gestión de archivos	45	84 %
Información del sistema	40	85 %
Control de procesos	35	83 %
Redes	30	93 %
Permisos	25	80 %
Total	200	89 %

de privilegios, modificaciones del sistema)—tasa garantizada por expresiones regulares, independiente del modelo. El detector local resolvió el 29 % de tareas sin LLM, con 100 % de precisión, ahorrando 500–1500 ms por solicitud.

4.2. Estudio piloto de usabilidad

Tres usuarios sin experiencia en GNU/Linux evaluaron BOTAS en su etapa de producto mínimo viable. Aunque insuficiente para generalización estadística, el piloto aportó hallazgos cualitativos: los participantes valoraron la ejecución por lenguaje natural, las explicaciones educativas y la retroalimentación de seguridad. La retroalimentación constructiva identificó confusión con la sintaxis verbal, frustración ante confirmaciones percibidas como excesivas y dificultad con mensajes de error—informando directamente el rediseño hacia diálogos de clarificación. Latencia: Whisper promedia 1–3 s; Vosk responde en <200 ms.

5. Discusión y Conclusiones

La precisión del 89 % implica que aproximadamente una de cada diez peticiones genera un comando incorrecto. Sin embargo, el diseño HCAI mitiga el impacto de este margen: cada comando se muestra al usuario antes de su ejecución (transparencia), las operaciones riesgosas requieren confirmación explícita (control humano) y el 100 % de operaciones destructivas son detectadas por la capa de seguridad determinista. Así, un comando incorrecto no se ejecuta automáticamente—el usuario conserva la decisión final, y el sistema explica cada acción propuesta, convirtiendo el error en oportunidad educativa. Desde la perspectiva latinoamericana, la barrera lingüística y los costos de ecosistemas cerrados representan obstáculos para instituciones educativas [Warschauer 2004]; la interacción por voz en español reduce estas barreras.

Trabajo futuro: consolidar la operación local, un estudio formal de usabilidad con muestra ampliada que evalúe comprensión, percepción de control, confianza y seguridad percibida, validación multidistribución y liberación como software libre. Los principios HCAI materializados en BOTAS demuestran que las barreras de acceso a sistemas potentes pueden reducirse sin sacrificar sus capacidades.

Agradecimientos

Este trabajo fue parcialmente desarrollado durante el Servicio Social en la Universidad Tecnológica de la Mixteca. El autor agradece al Dr. Ricardo Ruiz Rodríguez por su orien-

tación y apoyo a lo largo del desarrollo de esta investigación, así como a los participantes del estudio piloto de usabilidad por su retroalimentación.

Referencias

- Amershi, S., Weld, D., Vorvoreanu, M., Fournery, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., and Horvitz, E. (2019). Guidelines for human-AI interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33.
- Grudin, J. (2017). From tool to partner: The evolution of human-computer interaction. *Synthesis Lectures on Human-Centered Informatics*, 10(1):1–183.
- Jenson, S. (2024). The future of desktop UX: Breaking the cycle of stagnation. Invited talk, Akademy Conference. Available at <https://jenson.org>.
- Lin, X. V., Wang, C., Zettlemoyer, L., and Ernst, M. D. (2018). NL2Bash: A corpus and semantic parser for natural language interface to the Linux operating system. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*.
- Luger, E. and Sellen, A. (2016). “like having a really bad PA”: The gulf between user expectation and experience of conversational agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5286–5297.
- Norman, D. A. (1988). *The Design of Everyday Things*. Basic Books.
- OpenAI (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.
- Raymond, E. S. (2003). *The Art of UNIX Programming*. Addison-Wesley.
- Shneiderman, B. (2020). Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human-Computer Interaction*, 36(6):495–504.
- Shneiderman, B. (2022). *Human-Centered AI*. Oxford University Press.
- Top500.org (2023). Operating system family / Linux. <https://www.top500.org/statistics/details/osfam/1/>.
- W3Techs (2023). Usage statistics of Linux for websites. <https://w3techs.com/technologies/details/os-linux>.
- Warschauer, M. (2004). *Technology and Social Inclusion: Rethinking the Digital Divide*. MIT Press.