

Ficha de SOPHIA

Información general

Tipo de software	Aplicación local de escritorio en Python/Tkinter.
Origen	Iniciativa personal surgida de un flujo de traducción real, disperso entre múltiples tareas y herramientas, con numerosos cambios de contexto y una elevada carga de seguimiento mental.
Finalidad	Integrar el proceso de trabajo, automatizar tareas manuales y facilitar la orientación y la continuidad entre fases.
Desarrollo	2025–2026 · Desarrollo individual con asistencia de IA generativa.
Autoría	Érika García Bellés · Concepto, diseño, desarrollo, uso y pruebas.
Ámbito de aplicación	Automatización textual, traducción y control de calidad (QA).
Características de diseño	Arquitectura modular, motor operativo y andamiaje cognitivo integrado en el flujo de trabajo.
Idioma de la interfaz	Español e inglés. Terminología híbrida adaptada al flujo de trabajo original; sin versiones lingüísticas independientes.
Estado	Operativa en su entorno profesional original; no distribuida como producto comercial.
Transferibilidad	Principios de análisis y diseño de SOPHIA aplicables a otros procesos complejos.

Características y recursos

Dimensión operativa	Automatización de tareas, transformación de contenidos y preservación de datos y estructura a lo largo del proceso.
Dimensión cognitiva	Orientación contextual, seguimiento del progreso y recursos para reducir la dependencia de la memoria de trabajo.
Orientación y ayuda	Quick Start, instrucciones contextuales y ayudas integradas para facilitar la navegación, el avance y la continuidad del trabajo.
Memoria de traducción ligera	Consulta local de segmentos bilingües en un único archivo XLSX para recuperar traducciones previas y contexto.
Herramientas complementarias	Análisis SEO y extracción de fechas (Date Extractor).
Documentación	README por módulos y documentación estructurada con una fuente única de referencia (SSOT).

Información técnica

Lenguaje y entorno	Python · Tkinter · Aplicación de escritorio modular.
Arquitectura	Interfaz gráfica por pestañas, módulos independientes y scripts auxiliares para automatización y procesamiento por lotes.
Bibliotecas principales	openpyxl, pandas, regex, Beautiful Soup, PyTorch y Transformers (para OPUS-MT).
Tecnologías utilizadas	IA generativa (LLM/API), traducción automática neuronal (NMT/OPUS-MT) y AutoHotkey.
Papel de la IA generativa	Asistencia durante el desarrollo e integración de modelos de lenguaje mediante API en determinadas tareas de traducción, revisión y edición.
Compatibilidad TAO/CAT	Integración en flujos de trabajo con memoQ mediante preparación, conversión e intercambio de archivos.
Formatos de archivo	HTML, TXT, XLSX, JSON, MQXLIFF y MQXLZ, según el módulo.
Reglas de segmentación	Segmentación independiente de archivos TXT mediante reglas de memoQ (.mqres), con procesamiento individual o por lotes, exportación a XLSX y generación de un mapa de párrafos en JSON.