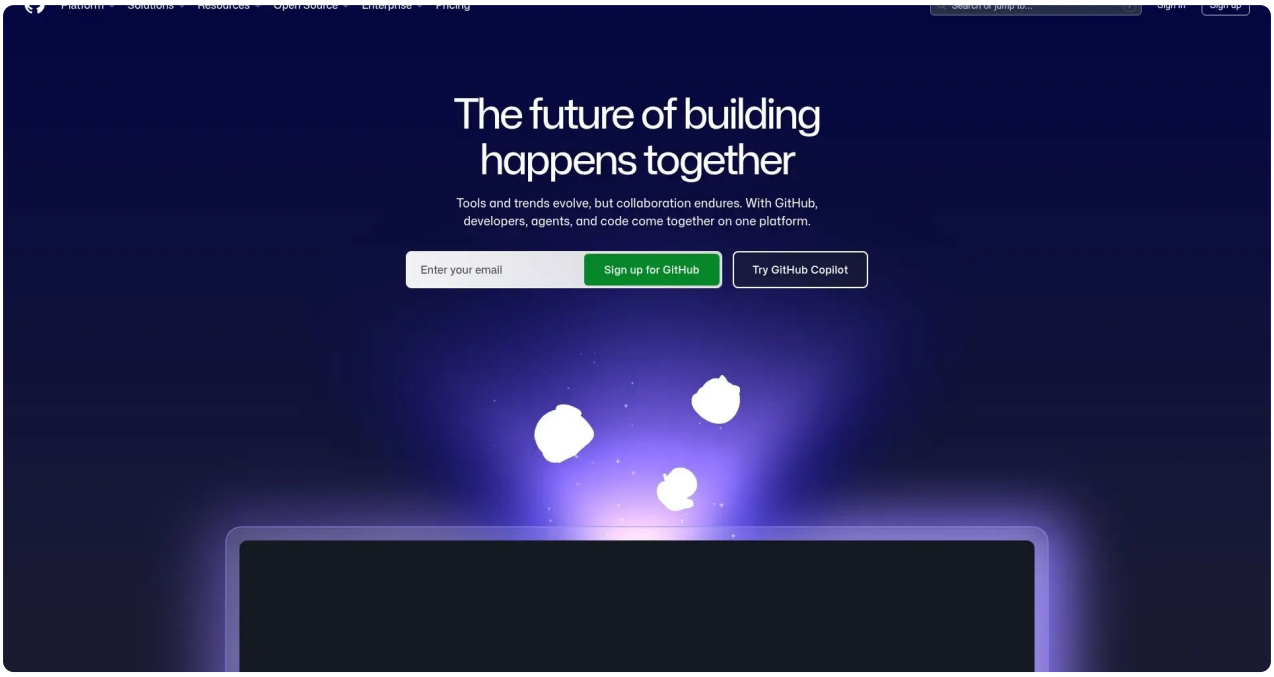

Agent S Framework

Agent S es un framework de agentes de IA de código abierto diseñado para desarrolladores, ingenieros de automatización e investigadores que necesitan operar sistemas operativos mediante visión y control de GUI. Utiliza Modelos de Lenguaje Multimodales para ver la pantalla, planificar tareas complejas y ejecutar acciones de ratón y teclado de forma autónoma. Es ideal para automatizar flujos de trabajo en cualquier software profesional, incluso sin APIs, superando el rendimiento humano en benchmarks.

Visitar Sitio Oficial Preguntar a ChatGPT Preguntar a Claude Preguntar a Grok

Contenido del Dossier

INFORMACIÓN DE LA HERRAMIENTA

Qué y para quién es
Agent S es un framework de agentes de IA de código abierto diseñado para operar un ordenador (GUI) de la misma forma que lo haría un humano. Utiliza Modelos de Lenguaje Multimodales (MLLM) para ver la pantalla, planificar tareas complejas en varios pasos y ejecutar acciones directas sobre el sistema operativo mediante ratón y teclado. Está dirigido a desarrolladores, ingenieros de automatización e investigadores de IA que buscan implementar flujos de trabajo autónomos que no dependen de APIs específicas, sino de la interfaz visual de las aplicaciones.

Principal ventaja profesional
Es el primer framework capaz de superar el rendimiento humano en el benchmark OSWorld (72.60%), permitiendo automatizar tareas profesionales en cualquier software (desde hojas de cálculo hasta herramientas de diseño o terminales) sin necesidad de que estos tengan integraciones previas.

Para quién no es
No es adecuado para usuarios finales sin conocimientos técnicos que busquen una solución "instalar y usar" con interfaz gráfica amigable, ni para entornos corporativos con restricciones estrictas de seguridad que prohíban la ejecución de código Python para el control del sistema.

Funcionalidades clave
- Interfaz Agente-Computador (ACI): Traduce intenciones de alto nivel en coordenadas y acciones de teclado/ratón sobre Windows, macOS y Linux.
- Planificación Jerárquica Aumentada por Experiencia: Divide tareas complejas en subtareas manejables utilizando memoria episódica y operativa.
- Recuperación de Conocimiento en Línea: Capacidad de buscar en la web (vía Perplexica) cómo usar software desconocido o actualizado.
- Entorno de Programación Local: Puede generar y ejecutar scripts de Python o Bash para procesar datos, manipular archivos o realizar configuraciones de sistema.
- Multimodalidad: Combina la visión de capturas de pantalla con el análisis del árbol de accesibilidad del sistema para una mayor precisión.

Precios
- Versión gratuita: El framework es Open Source bajo licencia Apache 2.0 (completa y gratuita para uso comercial o personal).
- Costes operativos: Requiere claves API de proveedores de LLM (OpenAI, Anthropic, Gemini) cuyos costes dependen del consumo de tokens y procesamiento de imágenes.
- Simular Cloud: Existe una opción en la nube (SaaS) proporcionada por los creadores para evitar la configuración local (consultar precios en la web oficial).

Perfil del usuario
- Ingenieros de Software y DevOps especializados en automatización de procesos complejos.
- Científicos de Datos e Investigadores de IA que necesiten herramientas de control de GUI.
- Responsables de QA para pruebas de software automatizadas en entornos reales.
- Departamentos de Operaciones que busquen automatizar tareas repetitivas en software legado sin API.

Nivel técnico requerido
- Nivel técnico de uso: Medio/Alto. Requiere interactuar mediante CLI o scripts de Python.
- Nivel técnico de instalación: Alto. Necesita configuración de entornos Python, manejo de variables de entorno y, opcionalmente, servidores OCR o Docker para funciones avanzadas.
- Competencias necesarias: Dominio de Python, gestión de claves API y familiaridad con sistemas operativos a nivel de permisos de control.

Ejemplos de uso profesional
- Gestión de correo: Abrir un cliente como Thunderbird, identificar cuentas específicas siguiendo instrucciones en lenguaje natural y eliminarlas o configurarlas.
- Procesamiento de datos: Abrir archivos Excel, realizar cálculos complejos mediante scripts locales y volcar los resultados en un CRM basado en web.
- Configuración de sistemas: Navegar por los menús de configuración del sistema operativo para aplicar

políticas de seguridad o ajustes de red.

Uso y distribución
- Versión web: Disponible mediante Simular Cloud.
- Versión escritorio: Compatible con Windows, macOS y Linux mediante instalación local.
- CLI: Interfaz de línea de comandos incluida para ejecución directa de agentes.

Open source
El proyecto es totalmente de código abierto, alojado en GitHub bajo licencia Apache License 2.0.

Integraciones
- Facilidad de integración: High Code (SDK en Python).
- API propia: Se distribuye como el paquete gui-agents instalable vía pip.
- Integraciones nativas: Soporta modelos de OpenAI (GPT-4o, o3), Anthropic (Claude 3.5/3.7), Google Gemini y modelos locales vía vLLM u OpenRouter. Se integra con Perplexica para búsqueda web y PaddleOCR para reconocimiento de texto en pantalla.

Notas finales
Información legal, licencias y contratos
- Licencia: Apache License 2.0. Permite uso, modificación y distribución comercial de forma gratuita.
- Privacidad: Al ser una ejecución local, el usuario tiene el control, pero las capturas de pantalla se envían a los proveedores de LLM configurados (OpenAI, etc.), lo que debe considerarse en términos de confidencialidad.

Otros
- Seguridad: El agente puede ejecutar código arbitrario y mover el cursor de forma autónoma. Se recomienda su uso en entornos aislados (Sandboxes) o máquinas dedicadas para evitar riesgos operativos.

Para más información:
- Sitio web oficial: https://www.simular.ai
- Github: https://github.com/simular-ai/agent-s
- Documentación técnica: https://docs.simular.ai/agent-s/introduction
- Publicación de investigación (Paper): https://arxiv.org/abs/2410.08164

CONSEJOS DE IMPLANTACIÓN

Aplicación profesional
Agent S es un framework de vanguardia para la automatización de interfaces gráficas (GUI) que permite a las empresas automatizar procesos en software donde no existen APIs (software legado, aplicaciones de escritorio locales, terminales). Es especialmente relevante para sectores con alta carga administrativa manual y procesos multi-paso.
- **Tipos de empresa:** Agencias de automatización, departamentos de IT/DevOps, empresas con software especializado (diseño, ingeniería) y sectores con sistemas heredados (Banca, Seguradoras).
- **Presupuesto:** El software es Open Source (Gratis). Los costes reales derivan del consumo de tokens de MLLM (GPT-4o, Claude 3.5 Sonnet) y la infraestructura de computación (servidores con capacidad de procesamiento de imagen).
- **Puntos clave:** Capacidad de aprendizaje mediante "Narrative Memory" y búsqueda en línea (vía Perplexica) para entender interfaces nuevas de forma autónoma.

Madurez digital requerida
- **Usuarios:** Nivel técnico alto. Requiere desarrolladores capaces de configurar entornos en Python, gestionar contenedores Docker y entender el flujo de agentes multimodales.
- **Empresa:** Debe contar con una infraestructura de datos y seguridad que permita la ejecución de scripts autónomos que controlan el ratón y el teclado en entornos controlados (Sandboxes).

Plan orientativo de implantación
Pasos necesarios y estimaciones
- **Tiempos estimados:** 4 a 8 semanas para un piloto funcional.
- **Evaluación inicial (Semana 1):** Identificación de flujos GUI críticos que carecen de API. Auditoría de compatibilidad de los sistemas operativos (Windows/Linux/macOS).
- **Configuración técnica (Semana 2):** Instalación del paquete gui-agents, configuración de claves API de MLLMs y despliegue de entornos aislados para pruebas de seguridad.
- **Prueba de concepto (Semanas 3-4):** Entrenamiento del agente en una tarea específica (ej. conciliación de facturas en software local). Configuración de la memoria episódica para que el agente "recuerde" los elementos visuales comunes.
- **Refinamiento y Human-in-the-loop (Semanas 5-8):** Implementación de capas de revisión donde el agente solicita aprobación antes de realizar acciones críticas (clics en "Enviar" o "Borrar").
- **Seguimiento:** Monitorización de la tasa de éxito (OSWorld benchmark como referencia interna) y ajuste de prompts de planificación jerárquica.

Perfiles necesarios
- **Ingenieros de Software / IA:** Para la lógica de orquestación y personalización del framework.
- **Especialistas en QA Automatizado:** Para diseñar los escenarios de prueba y validar que el agente actúa correctamente en la GUI.
- **Arquitectos de Seguridad:** Esencial para definir los límites de ejecución del agente y evitar riesgos de seguridad operativa.

Retorno de la inversión (ROI)
- **Tiempos:** Reducción del 60-80% en el tiempo de ejecución de tareas manuales repetitivas en el escritorio.
- **Cómo medirlo:**
- **Tasa de éxito de tarea:** Porcentaje de acciones completadas sin error humano.
- **Coste por ejecución:** Comparativa entre el coste de tokens vs. el coste por hora de un operario humano.
- **Latencia de proceso:** Tiempo total desde que se lanza el comando hasta que el agente cierra la aplicación.

Otros
- **Seguridad y Ética:** Dado que Agent S "ve" la pantalla, es crítico filtrar datos sensibles (PII) antes de enviarlos a los modelos de lenguaje en la nube (OpenAI/Anthropic). Se recomienda el uso de modelos locales para datos altamente confidenciales.
- **Limitaciones actuales:** Sensibilidad a cambios drásticos en la resolución de pantalla o actualizaciones de software que cambien radicalmente la disposición visual de los elementos.

PREGUNTAS FRECUENTES

¿Qué es Agent S y en qué se diferencia de un RPA tradicional?

Agent S es un framework de agentes de IA de código abierto diseñado para operar ordenadores de forma autónoma mediante la interfaz gráfica de usuario (GUI). A diferencia de las soluciones de Automatización Robótica de Procesos (RPA) tradicionales, que suelen basarse en reglas rígidas y selectores de código, Agent S utiliza Modelos de Lenguaje Multimodales (MLLM). Esto le permite ver la pantalla, razonar sobre el contenido visual y ejecutar acciones de teclado y ratón como lo haría un humano, siendo capaz de adaptarse a cambios en la interfaz y manejar tareas complejas sin necesidad de integraciones vía API.

¿Es Agent S una herramienta de código abierto?

Sí, el proyecto es totalmente Open Source y su código está disponible públicamente en GitHub. Se distribuye bajo la licencia Apache License 2.0, lo que permite a los profesionales y empresas utilizar, modificar y distribuir el software de forma gratuita, incluso para fines comerciales, sin restricciones de propiedad.

¿Cuáles son los costes asociados al uso de este framework?

Aunque el software es gratuito, su funcionamiento conlleva costes operativos derivados del uso de Modelos de Lenguaje de gran tamaño (LLM). El usuario debe proporcionar sus propias claves API de proveedores como OpenAI, Anthropic o Google Gemini, cuyos precios dependen del volumen de tokens procesados y la cantidad de capturas de pantalla analizadas. Adicionalmente, existe una opción gestionada en la nube denominada Simular Cloud con su propio modelo de suscripción para quienes prefieran evitar la gestión de la infraestructura local.

¿Cumple Agent S con los estándares de privacidad y seguridad corporativa?

La seguridad es un factor crítico ya que el agente puede ejecutar código arbitrario y controlar físicamente el ratón y teclado. Se recomienda estrictamente su ejecución en entornos aislados o 'Sandboxes'. Respecto a la privacidad, aunque el procesamiento puede ser local, el sistema envía capturas de pantalla a los proveedores de modelos (como OpenAI), por lo que la información sensible visible en pantalla podría ser procesada por terceros. Es responsabilidad del profesional configurar los filtros y entornos adecuados para cumplir con la normativa de protección de datos.

¿Qué nivel técnico se requiere para su implementación?

El nivel técnico requerido es alto. No es una solución orientada a usuarios finales sin conocimientos de programación. Para su instalación y uso profesional, es necesario tener experiencia en entornos Python, gestión de dependencias, configuración de variables de entorno y manejo de interfaces de línea de comandos (CLI). Su integración se realiza principalmente mediante el SDK `gui-agents` instalable vía pip.

¿Qué sistemas operativos y modelos de IA son compatibles?

El framework es compatible con Windows, macOS y Linux. En cuanto a la inteligencia artificial, soporta modelos punteros como GPT-4o, Claude 3.5/3.7 y Google Gemini. También ofrece flexibilidad para utilizar modelos locales a través de vLLM u OpenRouter, permitiendo una personalización técnica profunda según las necesidades del proyecto.

¿Cómo afronta el agente el uso de aplicaciones que no conoce?

Agent S incorpora una funcionalidad de 'Recuperación de Conocimiento en Línea' integrada con Perplexica. Esto le permite realizar búsquedas en tiempo real en la web para aprender a interactuar con software desconocido, consultar documentación actualizada o resolver dudas sobre flujos de trabajo específicos, superando la limitación de la fecha de corte del conocimiento del modelo base.

¿Es capaz de realizar tareas que requieran procesamiento de datos internos?

Sí, el framework incluye un entorno de programación local que le permite generar y ejecutar scripts en Python o Bash. Esto facilita el procesamiento de archivos locales, como hojas de cálculo o bases de datos, y la manipulación de información directamente en el sistema de archivos del host de forma complementaria a sus acciones en la interfaz visual.

CONTRATOS Y CONDICIONES

Informe técnico descriptivo

Principales recomendaciones
- **Ejecución en Sandbox**: Dado que la herramienta ejecuta código arbitrario (Python/Bash) y controla periféricos (ratón/teclado), es imperativo utilizar entornos aislados o máquinas virtuales para evitar daños accidentales en el sistema operativo local.
- **Supervisión Humana (Human-in-the-loop)**: No se recomienda delegar tareas críticas de forma 100% autónoma sin un mecanismo de validación, ya que las acciones sobre la interfaz gráfica pueden incurrir en errores si la ventana cambia de posición o surgen pop-ups inesperados.
- **Gestión de Secretos**: Evitar el uso de claves API de modelos de lenguaje (LLM) con límites de facturación abiertos; se deben configurar cuotas para prevenir costes imprevistos derivados de bucles infinitos en la ejecución del agente.
- **Revisión de Privacidad del Proveedor LLM**: Al capturar la pantalla para enviarla a modelos externos (GPT-4o, Claude 3.5), se debe asegurar que no haya información sensible visible en otras ventanas del escritorio.

Clasificación de impacto legal: ALTO
El impacto se considera alto debido a la capacidad de la herramienta para actuar en nombre del usuario, procesar datos visuales de cualquier aplicación (incluyendo datos sensibles de terceros) y ejecutar comandos de sistema, lo que activa múltiples obligaciones bajo el RGPD y la AI Act.

Ley de Inteligencia Artificial (AI Act)
- **Clasificación**: Se encuadra principalmente como un sistema de IA de propósito general con capacidades de agente. Si se utiliza para automatizar procesos en infraestructuras críticas o gestión de recursos humanos, podría ser clasificado como de **Alto Riesgo**.
- **Obligaciones de Transparencia**: La empresa debe informar claramente a cualquier tercero (empleados o clientes) que están interactuando con un sistema automatizado.
- **Control Humano**: La normativa exige que estos sistemas permitan una supervisión humana efectiva, lo cual es crítico dado que Agent S simula acciones humanas reales.

Privacidad y protección de datos
- **Responsabilidades**: La empresa usuaria actúa como **Responsable del Tratamiento**. Si se usa para procesar datos de clientes a través de interfaces gráficas, se debe realizar una Evaluación de Impacto en la Protección de Datos (EIPD).
- **Ubicación de los datos**: Aunque el framework corre localmente, el procesamiento visual se realiza mediante APIs de terceros (OpenAI en EE.UU., Anthropic, etc.). Esto implica una **Transferencia Internacional de Datos**.
- **Minimización**: El sistema captura la pantalla completa; es necesario configurar el entorno para que el agente solo "vea" la aplicación específica necesaria, evitando capturar notificaciones personales u otras apps abiertas.
- **Derechos ARCO**: La empresa debe garantizar que puede responder a solicitudes de acceso o supresión de los datos que el agente haya podido procesar o almacenar en su memoria episódica/narrativa.

Propiedad intelectual
- **Propiedad de datos**: Los "Prompts" y datos de entrada pertenecen al usuario. Según los términos de Simular Inc., los derechos sobre los "Outputs" (resultados) y "Actions" realizados son cedidos al usuario, siempre que se cumplan los términos de licencia.
- **Licencia del Framework**: Agent S se distribuye bajo **Apache License 2.0**, permitiendo el uso comercial, modificación y distribución sin coste de licencia, siempre que se mantengan los avisos de copyright.
- **Derivados**: El uso de esta tecnología para entrenar modelos de IA competidores está expresamente prohibido en los términos de servicio de la versión SaaS (Simular Pro).

Usos y prohibiciones
- **Usos admitidos**: Automatización de tareas administrativas, pruebas de software (QA), procesamiento de datos en software legado y flujos de trabajo multiapp.
- **Usos prohibidos**: Actividades que violen leyes locales o internacionales, scraping no autorizado de servicios protegidos, creación de productos competitivos con Simular Inc. mediante ingeniería inversa, y

entrega de asesoramiento profesional (médico, legal o financiero) basado únicamente en el agente.

Seguridad y certificaciones
- **Seguridad**: No se mencionan certificaciones SOC2 o ISO 27001 específicas para el framework de código abierto. La seguridad depende íntegramente de la infraestructura donde la empresa española despliegue el código.
- **Riesgo Específico**: El framework utiliza archivos .env para claves API; es crítico proteger estos archivos para evitar el robo de credenciales.

Otros
- **Memoria del Agente**: Agent S almacena "Memoria Episódica" y "Narrativa" para mejorar su rendimiento. El usuario debe auditar dónde se almacenan localmente estos archivos de registro para asegurar que no contienen datos personales persistentes sin protección.

Fuentes consultadas:
- Repositorio oficial Github
- Condiciones de servicio (Simular Pro / Terms of Service).docx.pdf)
- Documentación técnica oficial
- Política de privacidad y uso de datos
- Licencia Apache 2.0