

bitcompass refactor: extract duplicated window instance object into helper (#7938) 630385b · 19 hours ago · 7,803 Commits

- .devcontainer fix: update Tauri dependencies to use libappindicator3 instead... 2 months ago
- .github ci: temporarily disable portable artifact upload to GitHub rele... 3 weeks ago
- .husky chore: enhance onboarding screen's models (#4723) last year
- autoqa support 2 type of scroll behavior with proper settings (#6841) 6 months ago
- core fix: bypass auto-unload for local api server 2 months ago
- docs feat(mcp): routing telemetry, locale strings, and telemetry do... last week
- extensions fix: centralize Jan data folder path resolution (#7522) (#7929) 3 days ago
- flatpak fix: add Flatpak CUDA library path discovery and GL extensi... 5 days ago
- mix-server feat: add anthropic /messages endpoint last month
- scripts fix(#6902): update Bun download link for darwin-64x -> darw... 5 months ago
- src-tauri fix: remove core default permissions to unblock Tauri dev bul... 2 days ago
- tests support 2 type of scroll behavior with proper settings (#6841) 6 months ago
- web-app refactor: extract duplicated window instance object into help... 19 hours ago
- .gitattributes Create .gitattributes to suppress local Cargo.lock diffs (#7884) last week
- .gitignore Merge branch 'main' of https://github.com/janhq/jan into feat... 3 weeks ago
- .prettiignore fix: model path backward compatible (#2018) 2 years ago
- .prettierrc fix: model path backward compatible (#2018) 2 years ago
- .yamrc.yml chore: upgrade to turbo v2 and reduce ci quality gate runtim... 2 years ago

About
Jan is an open source alternative to ChatGPT that runs 100% offline on your computer.

jan.ai
open-source self-hosted gpt tauri
llm chatgpt llamapack localai

Releases 99
0.7.9 (Latest) 3 weeks ago
+ 98 releases

Packages
No packages published

Contributors 153

Jan AI

Jan es una plataforma de código abierto diseñada para profesionales, desarrolladores y empresas que necesitan ejecutar modelos de inteligencia artificial de forma 100% local y privada. Permite integrar LLMs como Llama o Mistral en flujos de trabajo corporativos sin depender de la nube, garantizando la soberanía del dato en sectores regulados como salud o finanzas. Es la herramienta ideal para quienes buscan un servidor API compatible con OpenAI pero operando totalmente offline en su propio hardware.

[Visitar Sitio Oficial](#) [Preguntar a ChatGPT](#) [Preguntar a Claude](#) [Preguntar a Grok](#)

Contenido del Dossier

- [Información de la Herramienta](#)
- [Consejos de Implantación](#)
- [Preguntas Frecuentes](#)
- [Contratos y Condiciones](#)

INFORMACIÓN DE LA HERRAMIENTA

Qué y para quién es

Jan es una alternativa de código abierto a ChatGPT diseñada para ejecutarse al 100% de forma local y privada en el hardware del usuario. Es una plataforma dirigida a profesionales, desarrolladores y empresas que buscan integrar Inteligencia Artificial en sus flujos de trabajo sin comprometer la privacidad de los datos ni depender de suscripciones a terceros. En el ámbito profesional, es ideal para sectores con estrictas normativas de cumplimiento (legal, salud, finanzas) y para departamentos técnicos que necesitan una infraestructura de IA personalizable y soberana.

Principal ventaja profesional

La **total privacidad y soberanía del dato**. Al procesar los modelos localmente (Local-first), la información sensible de la empresa nunca sale del equipo o servidor de la organización, eliminando riesgos de filtraciones en la nube y permitiendo trabajar de forma totalmente offline.

Para quién no es

No es apto para usuarios que carecen de hardware con potencia gráfica mínima (GPU) o RAM suficiente para mover modelos de lenguaje medianos. Tampoco es ideal para perfiles que buscan una solución "llave en mano" sin ninguna configuración o para aquellos que prefieren delegar toda la gestión de infraestructura y actualizaciones de modelos en un proveedor SaaS externo como OpenAI o Anthropic.

Funcionalidades clave

- **Inferencia Local de LLMs:** Permite descargar y ejecutar modelos (Llama, Gemma, Mistral, Qwen) directamente desde HuggingFace en formato GGUF.
- **Conectores Cloud:** Opción de conectar mediante API keys a modelos externos (OpenAI, Claude, Groq) si se requiere mayor potencia puntual.
- **Servidor API Compatible con OpenAI:** Expone un servidor local (localhost:1337) que permite integrar Jan con otras herramientas profesionales como VS Code, Continue o Cline.
- **Integración MCP (Model Context Protocol):** Conecta la IA con herramientas del mundo real (Canva, Jupyter, búsquedas web, bases de datos) para realizar tareas operativas.
- **Asistentes Personalizados:** Creación de hilos y roles específicos para tareas departamentales concretas.

Precios

- **Versión Gratuita:** Jan es una herramienta Open Source bajo licencia Apache 2.0. El software es completamente gratuito y funcional sin limitaciones para uso local.
- **Coste de Modelos Cloud:** Si se decide usar modelos externos (OpenAI/Anthropic), el usuario paga directamente al proveedor por su consumo de API; Jan no añade recargos.
- **Soporte Business:** Disponen de un canal de contacto para colaboraciones empresariales o necesidades específicas (hello@jan.ai).

Perfil del usuario

- **Empresas Tecnológicas:** Equipos de desarrollo que necesitan un copiloto de código privado.
- **Sectores Regulados:** Firmas legales, consultorías financieras y centros de salud que manejan datos sensibles.
- **Administraciones Públicas:** Organizaciones que requieren control total sobre el flujo de información.
- **Científicos de Datos:** Profesionales que evalúan modelos de código abierto en entornos controlados.

Nivel técnico requerido

- **Para uso básico:** Nivel usuario. La instalación es mediante un ejecutable (.exe, .dmg, .deb) con interfaz visual intuitiva.
- **Para configuración/desarrollo:** Nivel medio-alto. Requiere conocimientos sobre parámetros de modelos (context size, temperatura) y configuración de hardware (aceleración GPU).
- **Necesidades adicionales:** Dependencia del departamento de IT si se requiere desplegar la versión "Jan Server" en contenedores Docker para equipos.

Ejemplos de uso profesional

- **Análisis de documentos confidenciales:** Procesamiento de contratos o historiales médicos sin subir archivos a la red.
- **Asistencia en programación:** Integración como backend local para extensiones de autocompletado de

código.

- **Automatización de búsqueda interna:** Uso de protocolos MCP para consultar bases de datos locales mediante lenguaje natural.
- **Prototipado de IA:** Testeo de diferentes modelos Open Source para comparar rendimiento antes de un despliegue mayor.

Uso y distribución

- **Versión escritorio:** Disponible para Windows 10+, macOS 13.6+ (incluyendo Apple Silicon) y Linux (Ubuntu 20.04+).
- **CLI:** Herramienta de línea de comandos para gestión de servicios y configuraciones avanzadas.
- **Próximamente (Late 2025):** Versiones móviles (iOS/Android) y acceso vía web.

Open Source

El proyecto es totalmente de código abierto, distribuido principalmente bajo licencia Apache 2.0, lo que permite su auditoría, modificación y distribución comercial.

Integraciones

- **Facilidad de integración:** De No-code (interfaz de chat) a Full-code (uso de su API local).
- **API propia:** Servidor local compatible con el estándar de OpenAI, facilitando la sustitución de servicios de pago por el de Jan sin cambiar el código de las aplicaciones.
- **Protocolo MCP:** Soporta una amplia gama de conectores para interactuar con aplicaciones de terceros (Google Search, Slack, herramientas de diseño).
- **Docker:** Jan Server permite el despliegue mediante microservicios (Go, Docker Compose) para entornos corporativos escalables.

Notas finales

Información legal y licencias

Jan se distribuye bajo la licencia Apache 2.0. Los datos generados y los modelos descargados localmente son propiedad del usuario. Es importante revisar las políticas de privacidad individuales al conectar APIs de terceros (como OpenAI) a través de la interfaz de Jan.

Otros

Jan ofrece su propio hub de modelos optimizados (Jan-v1, Jan-Nano) diseñados específicamente para ofrecer buen rendimiento en hardware de consumo moderado.

Para más información:

- <https://jan.ai>
- <https://jan.ai/pricing>
- <https://github.com/janhq/jan>
- <https://discord.gg/jan-ai>

CONSEJOS DE IMPLANTACIÓN

Aplicación profesional

- **Sectores de alta privacidad:** Ideal para servicios jurídicos, salud, finanzas y administraciones públicas que procesan datos sensibles o confidenciales.
- **Desarrollo de software:** Equipos que requieren un "copiloto" de código privado integrado en VS Code o entornos corporativos cerrados.
- **Presupuesto:** Software gratuito (Open Source). El coste principal se deriva del **hardware** y el consumo eléctrico.
- **Puntos clave:** Ejecución 100% local, soberanía del dato, servidor API compatible con OpenAI y soporte del protocolo MCP para conectar la IA con archivos y bases de datos.

Madurez digital requerida

- **Usuarios:** Capacidad para gestionar archivos de modelos (GGUF), entender conceptos básicos de parámetros (temperatura, contexto) y configurar carpetas locales.
- **Empresa:** Infraestructura propia disponible para despliegue (GPUs NVIDIA o estaciones de trabajo con alta RAM) y protocolos internos de gobernanza de modelos de IA.

Plan orientativo de implantación

Pasos necesarios y estimaciones

- **Evaluación inicial (1-2 días):** Auditoría de hardware disponible. Se recomienda un mínimo de 16GB de RAM y GPUs NVIDIA (serie RTX 3000/4000) para un rendimiento fluido.
- **Prueba de concepto (1 día):** Instalación de la versión de escritorio y descarga de modelos ligeros (Llama 3 8B, Mistral) para validar que el hardware responde correctamente.
- **Configuración y Personalización (2-3 días):** Configuración del protocolo MCP para conectar la IA con el sistema de archivos de la empresa o bases de datos SQLite.
- **Despliegue Jan Server (3-5 días):** Si se requiere uso compartido, despliegue mediante Docker en servidores on-premise bajo Linux o Windows WSL2.
- **Capacitación (1-2 días):** Formación en el uso de asistentes específicos y gestión de hilos de chat.

Necesidades de formación del equipo

- Entrenamiento en "Prompt Engineering" específico para modelos locales (que suelen ser menos permisivos que GPT-4).
- Instrucción sobre la seguridad de los conectores Cloud (API keys) si se decide usar el modo híbrido.
- Gestión de la carpeta de datos y copias de seguridad de las configuraciones JSON.

Perfiles necesarios

- **Perfiles técnicos:** Administrador de sistemas con conocimientos en Docker (para Jan Server) y técnicos con experiencia en configuración de hardware GPU.
- **Personal externo:** Consultores en IA local para la optimización de modelos específicos según el caso de uso (quantization).

Retorno de la inversión (ROI)

- **Tiempos:** Ahorro inmediato en suscripciones recurrentes (SaaS) por usuario. Reducción de tiempos en procesos de análisis de documentos confidenciales que antes requerían revisión manual por seguridad.
- **KPIs:** Tokens por segundo (velocidad de respuesta), coste de hardware vs. suscripciones anuales, volumen de datos sensibles procesados sin salida a la nube.

Otros

- **Hardware Crítico:** Jan utiliza mucha más memoria de la que indican los modelos base debido al KV Cache y buffers. Para modelos de 8B, se recomiendan 12GB de VRAM real; para modelos de 70B, se requiere hardware profesional (A100/H100 o múltiples RTX 3090/4090).
- **Estabilidad:** Al ser un proyecto en desarrollo activo, se recomienda desactivar las actualizaciones automáticas en entornos de producción, ya que pueden resetear configuraciones personalizadas del archivo settings.json.
- **Protocolo MCP:** Permite que la IA ejecute código en notebooks de Jupyter o busque en la web, transformando el chat en una herramienta operativa, aunque requiere edición manual de archivos JSON para una configuración estable.

PREGUNTAS FRECUENTES

¿Qué es Jan y en qué se diferencia de ChatGPT?

Jan es una plataforma de inteligencia artificial de código abierto diseñada para ejecutarse localmente en el hardware del usuario. A diferencia de ChatGPT, que funciona en la nube, Jan permite procesar modelos de lenguaje sin conexión a internet, garantizando que los datos nunca salgan del dispositivo y eliminando la dependencia de suscripciones externas.

¿Es Jan una herramienta gratuita y de código abierto?

Sí, Jan es un software Open Source distribuido bajo la licencia Apache 2.0. Es completamente gratuito para descargar y utilizar de forma local tanto para usuarios individuales como para empresas. El código fuente está disponible públicamente en GitHub para su auditoría o modificación.

¿Cuáles son los requisitos de hardware necesarios para profesionales?

Para un rendimiento óptimo en entornos profesionales, se requiere hardware con una unidad de procesamiento gráfico (GPU) dedicada o memoria RAM suficiente para cargar los modelos de lenguaje. Es compatible con Windows 10+, macOS 13.6+ (optimizado para Apple Silicon) y distribuciones de Linux como Ubuntu 20.04+.

¿Cómo garantiza Jan la privacidad y el cumplimiento normativo?

Jan adopta un enfoque 'Local-first', lo que significa que la inferencia de los modelos y el almacenamiento de las conversaciones ocurren íntegramente en la infraestructura del usuario. Esto facilita el cumplimiento de normativas estrictas de protección de datos como el RGPD, ya que la información sensible no se transfiere a servidores de terceros.

¿Qué modelos de lenguaje se pueden ejecutar en la plataforma?

La plataforma permite descargar y ejecutar arquitecturas populares como Llama, Mistral, Gemma y Qwen desde HuggingFace en formato GGUF. Además, ofrece modelos propios optimizados como Jan-v1 y Jan-Nano para equipos con recursos de hardware moderados.

¿Es compatible con otras herramientas de desarrollo profesional?

Sí, Jan expone un servidor API local (puerto 1337) que es totalmente compatible con el estándar de OpenAI. Esto permite integrar la IA de Jan en flujos de trabajo existentes y herramientas de desarrollo como VS Code, Continue o Cline sin necesidad de reescribir código.

¿Permite Jan la conexión con modelos en la nube como OpenAI o Claude?

Aunque su foco es la ejecución local, Jan incluye conectores cloud opcionales. Los profesionales pueden vincular sus propias claves API de proveedores como OpenAI, Anthropic o Groq para realizar tareas que requieran una potencia de cómputo superior a la de su hardware local.

¿Qué es el protocolo MCP y cómo beneficia al usuario profesional?

El Model Context Protocol (MCP) permite a Jan interactuar con herramientas externas y fuentes de datos en tiempo real, como bases de datos, búsquedas web o aplicaciones de productividad. Esto transforma al asistente en un agente operativo capaz de ejecutar acciones y consultar información dinámica fuera de su entrenamiento estático.

¿Cómo se despliega Jan en un entorno empresarial con múltiples usuarios?

Para necesidades corporativas escalables, existe 'Jan Server', que puede ser desplegado mediante contenedores Docker. Esto permite a los departamentos de IT gestionar la IA como un microservicio dentro de la infraestructura privada de la empresa.

¿Requiere conexión a internet constante para funcionar?

No, una vez descargados los modelos de lenguaje, Jan funciona de manera totalmente offline. La conexión a internet solo es necesaria para la descarga inicial de modelos o si el usuario decide utilizar voluntariamente integraciones con APIs de terceros en la nube.

CONTRATOS Y CONDICIONES

Principales recomendaciones

- Evaluar el hardware corporativo: El uso profesional requiere GPUs dedicadas para garantizar la eficiencia. Un rendimiento lento puede comprometer la productividad.
- Establecer una política de "Proveedores Externos": Si se activan las extensiones de OpenAI, Anthropic o similares, los datos dejan de estar en local. Es crucial informar a los empleados de que solo el uso local garantiza la privacidad 100%.
- Auditoría de modelos: Al descargar modelos de HuggingFace, se debe verificar la licencia específica de cada modelo (ej. Llama 3, Mistral), ya que pueden tener restricciones comerciales diferentes a la licencia de Jan.
- Control de versiones: En despliegues mediante Docker para equipos, se recomienda centralizar la gestión para asegurar que todos los departamentos usen versiones validadas por IT.

Ley de Inteligencia Artificial (AI Act)

- Clasificación de riesgo: Jan es una herramienta de propósito general (GPAI). Al ejecutarse localmente y ser de código abierto (Apache 2.0), facilita el cumplimiento normativo al permitir auditorías totales del código.
- Obligaciones de transparencia: Si la empresa utiliza Jan para generar contenido que interactúa con terceros o clientes, debe informar de que el contenido ha sido generado por una IA, conforme al Reglamento de la UE.
- Uso en sectores críticos: Si se emplea en infraestructuras críticas o recursos humanos (riesgo alto), la empresa es la responsable de realizar la evaluación de impacto, no el desarrollador del software (Jan), al ser una implementación local self-hosted.

Privacidad y protección de datos

- Responsabilidades: La empresa española actúa como Responsable del Tratamiento y Encargado del Tratamiento simultáneamente. Al no haber un tercero (SaaS), la empresa tiene el control total sobre la seguridad de los datos.
- Ubicación de los datos: Los datos permanecen en los servidores o equipos de la empresa dentro del territorio español/UE. No existe transferencia internacional de datos en la configuración local.
- Transferencia internacional: Solo ocurre si el usuario configura manualmente API Keys de proveedores externos (EE. UU.). En ese caso, se requiere firmar Cláusulas Contractuales Tipo (SCC).
- Derechos ARCO: La empresa puede garantizar el acceso, rectificación, cancelación y oposición de forma inmediata puesto que posee el control físico de la base de datos de los chats (archivos JSON locales).

Propiedad intelectual

- Propiedad de datos: La empresa mantiene la propiedad absoluta sobre cualquier dato de entrenamiento, documentos subidos para contexto o prompts introducidos.
- Propiedad del resultado: Los resultados generados por la IA en entorno local pertenecen a la empresa, bajo los términos de la Ley de Propiedad Intelectual española, siempre que haya una intervención humana dirigida. Jan no reclama derechos sobre los outputs.

Usos y prohibiciones

- Usos prohibidos: No se debe utilizar para actividades que vulneren la privacidad de los empleados (vigilancia biométrica no autorizada) o para generar desinformación, conforme a las prohibiciones generales del AI Act.
- Usos admitidos: Análisis de contratos, asistencia en programación, clasificación de correos internos y cualquier tratamiento de datos sensibles (salud, religión, legal) que por ley no pueda subirse a nubes públicas.

Seguridad y certificaciones

- Seguridad: Al ser "Local-first", el riesgo de filtración por ciberataques a terceros desaparece. La seguridad depende exclusivamente del firewall y las políticas de ciberseguridad internas de la empresa española.
- Certificaciones: Jan es software de código abierto. La empresa puede auditar el código fuente en GitHub para verificar la ausencia de "backdoors" o telemetría no deseada.

Otros

- Licencia Apache 2.0: Esta licencia es muy permisiva y segura para empresas; permite modificar el código y usarlo con fines comerciales sin pagar royalties, siempre que se mantengan los avisos de copyright del autor original.
- Interoperabilidad: El servidor local compatible con la API de OpenAI facilita que la empresa migre de

soluciones en la nube a locales sin tener que reescribir sus aplicaciones internas de software.

Fuentes consultada:

- <https://jan.ai/docs>
- <https://github.com/janhq/jan/blob/main/LICENSE.md>
- <https://jan.ai/privacy>
- <https://jan.ai/terms>

Para más información y herramientas:

Explora look4.tools para descubrir las mejores soluciones tecnológicas del mercado.

[Inicio](#) [Todas las herramientas](#) [Categorías](#)

Este documento ofrece recomendaciones generadas mediante análisis humano y sistemas de IA automatizados. La información tiene carácter meramente informativo y no constituye asesoramiento legal, profesional ni garantía de resultados. Las marcas, logotipos y nombres comerciales pertenecen a sus respectivos propietarios y se utilizan únicamente con fines identificativos.