

# OpenCode AI Agent

*OpenCode es un agente de programación de código abierto diseñado para desarrolladores, ingenieros y arquitectos de software que buscan automatizar tareas complejas directamente desde su terminal o IDE. Permite conectar más de 75 proveedores de modelos de lenguaje, ofreciendo control total sobre la privacidad y la soberanía de datos. Es ideal para refactorización masiva, diagnóstico de errores y generación de tests, permitiendo el uso de modelos locales para entornos de máxima seguridad.*

[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

OpenCode es un agente de programación de código abierto diseñado para automatizar y asistir en tareas de desarrollo directamente desde la terminal, IDE o una aplicación de escritorio. A diferencia de soluciones cerradas, permite conectar más de 75 proveedores de modelos de lenguaje (LLM). Está dirigido a desarrolladores, ingenieros de software y arquitectos de sistemas que buscan una herramienta de alta productividad con control total sobre la privacidad y la elección de la inteligencia artificial.

### Principal ventaja profesional

La libertad absoluta de modelos y la soberanía de datos: permite realizar tareas complejas de codificación (refactorización, creación de archivos, ejecución de tests) utilizando cualquier modelo del mercado (Claude, GPT, Gemini) o modelos locales (Ollama), garantizando que el código no sale del entorno del desarrollador si así se configura.

### Para quién no es

No es para perfiles no técnicos o "no-coders" que buscan una interfaz puramente visual y guiada. Profesionales que prefieren ecosistemas cerrados y 100% asistidos sin necesidad de configurar APIs o entornos de ejecución locales pueden encontrar la herramienta excesivamente técnica.

### Funcionalidades clave

- Agente multi-sesión: Permite ejecutar múltiples agentes en paralelo sobre el mismo proyecto.
- Modos de trabajo: Incluye modo "Build" para ejecución completa de cambios y modo "Plan" para análisis y propuestas sin modificar archivos.
- Interfaz TUI avanzada: Terminal de usuario con scroll libre, redimensionado dinámico y resaltado de sintaxis integrado.
- Soporte LSP: Detecta y carga automáticamente los Language Server Protocols adecuados para mejorar la precisión del modelo.
- Análisis Air-gapped: Posibilidad de funcionamiento 100% offline mediante el uso de modelos locales.
- Compartición de sesiones: Generación de enlaces para debugear o revisar sesiones de codificación con otros compañeros.

### Precios

- Versión Open Source: Gratuita bajo licencia MIT. El usuario puede descargar, modificar y usar la herramienta sin coste, pagando únicamente por el consumo de las APIs de IA que decida conectar.
- OpenCode Zen: Modelo de pago por uso (pago previo de aprox. 20€ de saldo) que ofrece acceso a un catálogo de modelos optimizados y probados específicamente para programación, sin necesidad de gestionar múltiples suscripciones.
- OpenCode Enterprise: Modelo basado en licencias por puesto (per-seat) para organizaciones que requieren integración con SSO (Single Sign-On), configuración centralizada y pasarelas de IA internas.

### Perfil del usuario

- Desarrolladores Senior y Lead que gestionan arquitecturas complejas y requieren automatización en la terminal.
- Departamentos de Ciberseguridad que necesitan herramientas de IA que cumplan con normativas de privacidad estrictas (modelos locales).
- Equipos de DevOps para la automatización de scripts, parches y despliegues mediante CLI.

### Nivel técnico requerido

- Nivel de uso: Medio-Alto (familiaridad con la terminal, comandos shell y gestión de APIs).
- Nivel de instalación: Medio (instalación vía script de bash o binarios de escritorio; configuración de archivos JSON para personalización).
- Conocimientos necesarios: Manejo de Git, conocimiento de entornos de ejecución (Node/Bun para plugins) y conceptos básicos de LLMs (tokens, context window).

### Ejemplos de uso profesional

- Refactorización masiva de código: Cambiar patrones de diseño en todo un repositorio de forma automática.
- Diagnóstico y Debugging: Pedir al agente que analice logs de error y proponga o aplique la solución directamente.

- Generación de documentación y tests: Creación automática de archivos README o suites de testing unitario analizando la lógica existente.
- Planificación de proyectos: Uso del modo "Plan" para obtener un desglose de tareas antes de realizar cualquier cambio en el código fuente.

#### Uso y distribución

- Versión web (limitada a la gestión de sesiones compartidas).
- Extensión del navegador: Disponible para Chrome.
- Versión escritorio: Aplicación nativa para Windows, macOS (Apple Silicon y x64) y Linux.
- Extensiones de IDE: Soporte para VS Code, JetBrains y Cursor.
- CLI: Interfaz de línea de comandos robusta para uso directo en terminal.

#### Open Source

La herramienta es de código abierto bajo licencia MIT, con un repositorio muy activo y transparente en GitHub.

#### Integraciones

- Facilidad de integración: Alta (permite extender funcionalidades mediante plugins propios en JavaScript/TypeScript).
- API propia: Dispone de SDK para construir agentes personalizados y hooks de ciclo de vida.
- MCP (Model Context Protocol): Soporta servidores MCP nativos para ampliar el contexto del agente con herramientas externas.
- Integraciones nativas: GitHub Copilot, OpenAI, Anthropic, Google Gemini junto a más de 75 proveedores a través de Models.dev.

#### Notas finales

##### Información legal, licencias y contratos

OpenCode utiliza una licencia MIT, lo que permite su uso comercial, modificación y distribución con mínimas restricciones. El usuario mantiene la propiedad total de todo el código generado por la herramienta. En la versión Enterprise, se ofrecen contratos específicos de nivel de servicio (SLA) y soporte dedicado.

#### Otros

Es importante destacar el soporte nativo para el entorno Bun y Zig en su arquitectura, lo que le confiere una velocidad de respuesta muy superior a otros agentes basados íntegramente en Electron.

#### Para más información:

- Sitio web oficial: <https://opencode.ai>
- Documentación técnica: <https://opencode.ai/docs>
- Precios (Zen): <https://opencode.ai/es/zen>
- Precios (Enterprise): <https://opencode.ai/es/enterprise>
- Github: <https://github.com/opencode-ai/opencode>
- Twitter / X: <https://x.com/opencodeai>

## CONSEJOS DE IMPLANTACIÓN

---

### Aplicación profesional

- Dirigido a departamentos de desarrollo de software, consultoras tecnológicas y empresas con altos estándares de privacidad de datos.
- Presupuesto: Bajo para la versión Open Source (solo costes de cómputo/API), medio para escalado empresarial (licenciamiento por puesto).
- Puntos clave: Reducción del tiempo de desarrollo en tareas repetitivas, estandarización de código mediante refactorizaciones masivas y seguridad total a través de modelos locales (Air-gapped).

### Madurez digital requerida

- Usuarios: Desarrolladores senior con dominio de terminal (CLI), gestión de Git y configuración de entornos de ejecución (Node.js/Bun).
- Empresa: Organizaciones con cultura DevOps establecida y flujos de trabajo basados en revisión de código y testing automatizado.

### Plan orientativo de implantación

#### Pasos necesarios y estimaciones

- Tiempos estimados de despliegue: De 1 a 3 semanas dependiendo de la complejidad de la infraestructura de IA elegida.
- Evaluación inicial: Definición de proveedores de LLM (locales como Ollama o externos como Anthropic/OpenAI) y auditoría de seguridad sobre la salida de datos.
- Implantación inicial: Configuración de la CLI y el entorno de escritorio en un grupo de control (2-5 desarrolladores) para validar la integración con el stack tecnológico de la empresa.
- Prueba de concepto: Ejecución de tareas de refactorización y documentación automática sobre un repositorio no crítico.
- Formación y adaptación: Sesiones técnicas sobre el uso de los modos "Build" y "Plan", y configuración del Model Context Protocol (MCP) para conectar bases de datos o documentación interna.
- Seguimiento: Evaluación de la calidad del código generado y ajuste de los "system prompts" o modelos utilizados según los resultados obtenidos.

### Necesidades de formación del equipo

- Configuración de servidores MCP para ampliar el contexto del agente.
- Gestión de costes de tokens y optimización de contextos en modelos comerciales.
- Operativa de modelos locales (Ollama/Llama.cpp) para tareas de máxima privacidad.

### Perfiles necesarios

- Perfiles técnicos: Desarrolladores Senior, Arquitectos de Software y especialistas en DevOps/SRE.
- Personal externo: Consultores en IA generativa solo si se requiere una integración compleja de modelos locales en servidores corporativos.

### Retorno de la inversión

- Tiempos: Se estima una reducción del 30-40% en tiempos de generación de tests unitarios y documentación técnica desde el primer mes.
- Cómo medirlo: KPIs basados en la reducción de deuda técnica, aumento en la frecuencia de commits de calidad y disminución del tiempo medio de resolución de bugs (MTTR).

### Otros

- Rendimiento: Al estar construido con soporte nativo para Bun y Zig, ofrece una latencia significativamente menor que herramientas basadas exclusivamente en Electron, facilitando un flujo de trabajo más ágil en máquinas locales con recursos limitados.
- Flexibilidad de modelos: La compatibilidad con más de 75 proveedores evita el "vendor lock-in", permitiendo cambiar de modelo de IA instantáneamente según la evolución del mercado o cambios en los precios de las APIs.

## PREGUNTAS FRECUENTES

---

### ¿Qué es OpenCode y a quién está dirigido?

OpenCode es un agente de programación de código abierto diseñado para automatizar y asistir en tareas de desarrollo de software desde la terminal, IDE o escritorio. Está orientado específicamente a perfiles técnicos como desarrolladores senior, ingenieros de software y arquitectos de sistemas que requieren un control granular sobre sus herramientas y la privacidad de su código.

### ¿Es OpenCode una herramienta gratuita y de código abierto?

Sí, OpenCode es open source y se distribuye bajo la licencia MIT, lo que permite su descarga, modificación y uso comercial sin coste. El código fuente es accesible públicamente en GitHub, fomentando la transparencia y la colaboración de la comunidad.

### ¿Cómo garantiza la privacidad y la seguridad de los datos?

La herramienta permite una soberanía total de los datos mediante la configuración de modelos en modo 'Air-gapped' u offline. Al soportar integraciones locales como Ollama, el código fuente no necesita salir del entorno del desarrollador, cumpliendo así con normativas estrictas de seguridad de la información.

### ¿Qué modelos de inteligencia artificial se pueden utilizar?

Ofrece compatibilidad con más de 75 proveedores de modelos de lenguaje (LLM). Esto incluye modelos comerciales destacados como Claude (Anthropic), GPT (OpenAI) y Gemini (Google), además de la posibilidad de conectar modelos locales de código abierto.

### ¿Cuáles son las diferencias entre las versiones Open Source, Zen y Enterprise?

La versión Open Source es gratuita y el usuario gestiona sus propias APIs. OpenCode Zen es un modelo de pago por uso orientado a simplificar el acceso a modelos optimizados sin múltiples suscripciones. OpenCode Enterprise está diseñado para organizaciones, ofreciendo licencias por puesto, soporte para SSO, configuración centralizada y garantías de nivel de servicio (SLA).

### ¿Qué nivel de conocimientos técnicos se requiere para su implementación?

Se requiere un nivel técnico medio-alto. El usuario debe estar familiarizado con el uso de la terminal (CLI), gestión de comandos shell, manejo de Git y configuración de archivos JSON. No es una solución recomendada para perfiles 'no-coders' debido a su naturaleza técnica y requisitos de configuración de entornos.

### ¿En qué entornos y sistemas operativos se puede ejecutar?

Es multiplataforma y está disponible como aplicación nativa para Windows, macOS (Intel y Apple Silicon) y Linux. Además, cuenta con una interfaz de línea de comandos (CLI), extensiones para navegadores como Chrome e integraciones con los principales IDEs como VS Code y JetBrains.

### ¿Qué es el modo 'Plan' y en qué se diferencia del modo 'Build'?

El modo 'Plan' permite al agente realizar análisis, diagnósticos y propuestas de cambios sin modificar los archivos del proyecto, ideal para la fase de diseño. El modo 'Build' otorga permisos para ejecutar cambios directos en el código, realizar refactorizaciones y ejecutar suites de tests de manera autónoma.

### ¿Soporta el protocolo MCP (Model Context Protocol)?

Sí, OpenCode tiene soporte nativo para servidores MCP. Esto permite ampliar el contexto y las capacidades del agente conectándolo con herramientas y fuentes de datos externas, mejorando la precisión en la resolución de tareas complejas.

### ¿Es posible extender las funcionalidades de la herramienta?

Sí, la plataforma es altamente extensible. Permite a los desarrolladores crear sus propios plugins utilizando JavaScript o TypeScript mediante un SDK propio y hooks de ciclo de vida definidos en su arquitectura.

## CONTRATOS Y CONDICIONES

### Principales recomendaciones

- Configuración de soberanía: Para cumplir con políticas de seguridad corporativa, priorice el uso de modelos locales (Ollama o similares) o instancias privadas en la nube para evitar que el código fuente de la empresa sea procesado por servidores de terceros.
- Gestión de API Keys: En la versión Open Source y Zen, las claves de API se almacenan localmente; asegure el cifrado de los archivos de configuración JSON mediante herramientas de gestión de secretos corporativos.
- Auditoría de cambios: Al utilizar el modo "Build", implemente una revisión obligatoria de código (Code Review) antes de integrar los cambios sugeridos por el agente en ramas de producción, para evitar vulnerabilidades inyectadas accidentalmente.
- Debida diligencia en proveedores: Si opta por la versión Zen, verifique que los modelos LLM seleccionados dentro del catálogo cumplen con los requisitos de privacidad de su organización, ya que cada proveedor (OpenAI, Anthropic, etc.) tiene términos distintos.

### Ley de Inteligencia Artificial (AI Act)

- Clasificación de riesgo: Esta herramienta se clasifica generalmente como de "riesgo mínimo/bajo" al ser un sistema de asistencia a la programación. No obstante, si se utiliza para generar código que gestione infraestructuras críticas o sistemas de salud, el usuario final debe realizar una evaluación de impacto sectorial.
- Transparencia: El usuario debe ser consciente de que el código es generado por una IA. El uso de licencias MIT facilita la trazabilidad del software, cumpliendo con los requisitos de transparencia para sistemas de propósito general (GPAI).

### Privacidad y protección de datos

- Responsabilidades: La empresa usuaria actúa como Responsable del Tratamiento. OpenCode, en su versión local/Open Source, actúa solo como software, por lo que el control de los datos es total para el usuario.
- Ubicación de los datos: En el análisis "Air-gapped", los datos no salen de la infraestructura de la empresa. En el uso de APIs externas (Cloud), la ubicación dependerá del proveedor del LLM (habitualmente EE.UU.).
- Transferencia internacional: El uso de modelos externos puede implicar transferencias de datos fuera del Espacio Económico Europeo. Se recomienda el uso de modelos con servidores en la UE o la firma de Cláusulas Contractuales Tipo si se usan proveedores estadounidenses.
- Derechos ARCO: Dado que la herramienta puede procesar datos personales presentes en el código (nombres en comentarios, variables), la empresa debe garantizar que puede localizar y suprimir dicha información si un interesado lo solicita.

### Propiedad intelectual

- Propiedad de datos: El código fuente introducido en la herramienta pertenece exclusivamente a la empresa usuaria.
- Propiedad del resultado: Los términos de OpenCode estipulan que el usuario mantiene la propiedad total del código generado. Bajo la legislación española, el software generado puramente por IA no tiene autoría humana protegible por derechos de autor, por lo que se recomienda la intervención y validación humana para consolidar la protección jurídica del software resultante.

### Usos y prohibiciones

- Usos admitidos: Refactorización, documentación, generación de tests, diagnóstico de bugs y planificación de arquitectura técnica.
- Usos prohibidos: No debe utilizarse para la creación de código malicioso, bypass de medidas de seguridad o cualquier actividad que infrinja la Ley de Propiedad Intelectual de terceros mediante el entrenamiento no autorizado de modelos (si bien esto último depende del modelo de IA conectado, no de OpenCode).

### Seguridad y certificaciones

- Seguridad: La herramienta soporta arquitecturas desconectadas de internet (offline), eliminando el riesgo de filtración de datos exfiltration por parte del agente.
- Certificaciones: La versión Enterprise facilita la conformidad con normativas SOC2 o ISO 27001 al permitir el despliegue en entornos controlados y el uso de SSO (Single Sign-On).

### Otros

- Licencia MIT: Permite una libertad casi total para la empresa, incluyendo la modificación del core de la herramienta para adaptarlo a flujos de trabajo internos de cumplimiento normativo.
- Modelo Zen: El uso de saldo prepago simplifica la gestión administrativa y fiscal, evitando múltiples facturas de diferentes proveedores de IA.

Fuentes consultada:

- <https://github.com/opencode-ai/opencode>
- <https://opencode.ai/docs>
- <https://opencode.ai/es/enterprise>
- <https://opencode.ai/es/zen>
- <https://opensource.org/license/mit>

### 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.