



LangGraph

LangGraph es un framework de orquestación de bajo nivel diseñado para construir agentes de IA con estados complejos, ciclos y lógica personalizada. Permite a ingenieros de software y arquitectos de IA crear sistemas autónomos robustos mediante grafos donde los nodos son funciones y las aristas definen el flujo. Es ideal para profesionales que necesitan control total sobre el bucle de razonamiento, persistencia de memoria y supervisión humana en procesos críticos como finanzas o legal.

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

Contenido del Dossier

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

INFORMACIÓN DE LA HERRAMIENTA

Qué y para quién es

LangGraph es un framework de orquestación de bajo nivel diseñado para construir agentes de IA con estados complejos, ciclos y lógica personalizada. A diferencia de las cadenas lineales simples, permite crear grafos donde los nodos son funciones (o LLMs) y las aristas definen el flujo de control, permitiendo que el agente retroceda, itere o se detenga.

Está dirigido a ingenieros de software, arquitectos de IA y equipos de datos en empresas que han superado las limitaciones de las aplicaciones de "un solo paso" y necesitan sistemas autónomos robustos. Es ideal para sectores como el legal, financiero o de atención al cliente, donde los procesos requieren verificación, corrección de errores y memoria persistente.

Principal ventaja profesional

En mi opinión profesional, la razón definitiva para adoptar LangGraph es el control total sobre el "bucle de razonamiento". Mientras que otras herramientas ocultan la lógica en una "caja negra", LangGraph permite definir exactamente cuándo un agente debe pedir ayuda humana, cuándo debe reintentar una tarea o cómo debe actualizar su memoria a largo plazo. Es la diferencia entre un prototipo frágil y un sistema de producción predecible.

Para quién no es

No es para profesionales que buscan soluciones "no-code" o "click-and-run"; para ellos existen alternativas como LangFlow. Al probarlo, he verificado que requiere una mentalidad de desarrollador sólida. Equipos que solo necesiten una interfaz sencilla para chatear con un PDF encontrarán la curva de aprendizaje de LangGraph innecesariamente alta y optarán por las APIs directas de OpenAI o la capa superior de LangChain.

Funcionalidades clave

- Persistencia y Checkpointing: Permite guardar el estado del agente en cada paso, lo que facilita retomar tareas tras un error o una pausa larga.
- Human-in-the-loop: Funcionalidad nativa para detener la ejecución, esperar la aprobación o edición manual de un humano y continuar.
- Soporte de Ciclos: Capacidad real de iteración (re-planificación, re-ejecución) que no causa errores de recursividad infinita gracias a su gestión de estados.
- Memoria Multinivel: Gestión de memoria a corto plazo (sesión) y a largo plazo (entre diferentes conversaciones) integrada en el grafo.
- Streaming Fino: Capacidad de transmitir actualizaciones de estado nodo por nodo, ideal para interfaces de usuario dinámicas.

Precios

- Versión gratuita: La librería central de LangGraph es Open Source bajo licencia MIT (gratis para uso comercial y personal).
- Rango de precios adicionales: Basado en el uso de la plataforma LangSmith para despliegue y observabilidad.
- Plan Developer: 0€ (incluye hasta 5k trazas/mes).
- Plan Plus: Desde 39€ al mes por asiento (incluye despliegue gestionado y soporte por email).
- Plan Enterprise: Precios personalizados (para hosting híbrido, VPC y soporte dedicado).

Perfil del usuario

- Empresas de tecnología que integran agentes en sus propios productos (SaaS).
- Departamentos de Operaciones que automatizan flujos de trabajo complejos con múltiples herramientas externas.
- Desarrolladores de Backend y ML Engineers enfocados en la fiabilidad del software.

Nivel técnico requerido

- Nivel técnico para su uso: Alto. Requiere dominio de Python o TypeScript.
- Nivel técnico para configuración: Medio-Alto. La instalación es sencilla (`pip install langgraph`), pero el diseño de la arquitectura del grafo requiere conocimientos de lógica de programación y flujos de datos.
- Competencias necesarias: Conocimiento de decoradores, tipos de datos (TypedDict/Pydantic) y progra-

mación asíncrona.

Ejemplos de uso profesional

- Agentes de Atención al Cliente: Un flujo que intenta resolver la duda solo; si no puede, formatea el caso y se detiene automáticamente para que un agente humano valide la respuesta antes de enviarla.
- Pipelines de Extracción de Datos: Un sistema que descarga documentos, extrae campos y, si el LLM detecta una inconsistencia, vuelve a ejecutar el paso de búsqueda con nuevos parámetros.
- Generación de Código/Informes: Un agente que escribe una primera versión, la pasa por un "nodo revisor" que busca errores y lo devuelve al "nodo escritor" para correcciones hasta que cumpla el estándar de calidad.

Uso y distribución

- Versión web: Disponible a través de LangGraph Studio (interfaz visual para depurar grafos).
- Librerías: Paquetes oficiales para Python y TypeScript.
- Despliegue: Se puede auto-alojar en cualquier infraestructura (Docker) o usar LangGraph Cloud para despliegue gestionado.

Open source

La librería es Open Source (licencia MIT), lo que garantiza libertad de modificación y evita el bloqueo con un solo proveedor.

Integraciones

- Facilidad de integración: Full code. Se requiere código para conectar componentes, aunque se integra nativamente con todo el ecosistema LangChain.
- API propia: Los grafos compilados pueden exponerse como endpoints REST mediante LangGraph Server.
- Dispone de servidor MCP para conectar herramientas y modelos de forma estandarizada.
- Integración nativa total con LangSmith para monitorización y LangChain para el uso de cientos de modelos (OpenAI, Anthropic, Ollama, etc.) y bases de datos vectoriales.

Notas finales

Veredicto técnico

LangGraph es la herramienta de mayor utilidad actual para convertir agentes experimentales en aplicaciones profesionales. Vale totalmente la pena para empresas que ya tienen casos de uso de IA validados y sufren por la falta de control de las herramientas más simples. Como profesional valoro especialmente su capacidad de persistencia, que soluciona el problema de los agentes que "se pierden" en tareas largas.

Información legal

Licencia MIT para la librería. La propiedad intelectual de los grafos desarrollados pertenece íntegramente al usuario. Los datos enviados a LangSmith para observabilidad no se utilizan para entrenar modelos, cumpliendo estándares SOC2 y GDPR.

Otros

Quiero destacar que LangGraph ha sido adoptado por empresas como Uber y J.P. Morgan, lo que demuestra su madurez para entornos de alta seguridad y escalabilidad.

Fuentes consultadas:

- <https://www.langchain.com/langgraph>
- <https://docs.langchain.com/langgraph>
- <https://github.com/langchain-ai/langgraph>
- <https://www.langchain.com/langgraph-platform-pricing>

CONSEJOS DE IMPLANTACIÓN

Aplicación profesional

Según mi experiencia, LangGraph es la herramienta definitiva para empresas de servicios IT, departamentos de I+D y entornos corporativos (Fintech, Legaltech, Healthtech) que necesitan superar el estado de "demo" de la IA generativa. Lo que más me gusta es que permite construir sistemas que no solo generan texto, sino que ejecutan flujos de trabajo con lógica de negocio estricta. El presupuesto necesario es moderado en términos de licencias (puedes empezar a coste cero con la librería Open Source), pero elevado en capital humano, ya que requiere desarrolladores experimentados. En mi opinión profesional, es la inversión más rentable para empresas que buscan fiabilidad operativa y no solo innovación cosmética.

Madurez digital requerida

- **Usuarios y equipo:** Se requiere un equipo de desarrollo con dominio profundo de Python o TypeScript, familiaridad con arquitecturas orientadas a eventos y, preferiblemente, experiencia previa con LangChain. No es apto para equipos sin capacidad de programación backend.

- **Empresa y departamentos:** La organización debe contar con una infraestructura de datos mínima y una cultura de observabilidad de software. Es ideal para departamentos que ya han experimentado con LLMs y han chocado con la barrera de la alucinación o la falta de control en flujos complejos.

Plan orientativo de implantación

Pasos necesarios y estimaciones

- **Evaluación inicial (1-2 semanas):** Auditoría de los flujos de trabajo actuales. Identificación de puntos de decisión donde el agente de IA requiere supervisión humana o iteración. Definición de los estados del grafo.

- **Prueba de concepto y Diseño de Grafo (2-4 semanas):** Configuración del entorno de desarrollo y diseño del esquema de estado (State). Creación de un MVP que resuelva un cuello de botella específico, utilizando LangGraph Studio para la visualización del flujo.

- **Implantación y configuración (3-5 semanas):** Integración con sistemas existentes (APIs, bases de datos). Configuración de la persistencia mediante checkpoints en bases de datos (PostgreSQL/Redis) para garantizar la continuidad de las sesiones.

- **Piloto y Human-in-the-loop (2-4 semanas):** Despliegue en un entorno controlado donde expertos humanos validen las decisiones del agente antes de que tengan efectos reales.

- **Puesta en producción y Seguimiento (Continuo):** Monitorización a través de LangSmith para identificar cuellos de botella en los nodos del grafo y ajustar la lógica de las aristas.

Necesidades de formación del equipo

El equipo debe formarse específicamente en la gestión de estados persistentes y en el patrón de diseño de "Grafos Cíclicos Dirigidos". Es crucial entender cómo manejar la recursividad y los límites de pasos para evitar bucles infinitos que disparen los costes de cómputo.

Perfiles necesarios

- **Perfiles técnicos necesarios:** Ingenieros de IA/ML, Desarrolladores Backend Senior, Arquitectos de Soluciones.

- **Personal externo recomendado:** Consultores expertos en orquestación de LLM para la fase inicial de arquitectura.

- **Otros:** Analistas de negocio que puedan definir con precisión las reglas de decisión de los nodos del grafo.

Retorno de la inversión

- **Tiempos:** Se observa una reducción significativa en el tiempo de corrección manual de errores tras 3-6 meses de implantación.

- **Cómo medirlo, KPIs:** Tasa de éxito de las tareas completadas sin intervención humana frente a intervención manual (Success Rate), reducción de tokens desperdiciados por reintentos fallidos, y tiempo medio de resolución de procesos complejos (MTTR).

Otros

Al usarlo te das cuenta de que la mayor ventaja no es la IA en sí, sino la capacidad de "pausar" el estado del mundo. En mi experiencia en implantaciones, la funcionalidad de Checkpointing es lo que realmente permite escalar: si una API externa falla o el usuario se desconecta, el agente no olvida lo que estaba haciendo. Mi experiencia me lleva a pensar que LangGraph se convertirá en el estándar para el desarrollo de agentes autónomos empresariales debido a su capacidad de integrarse con el protocolo MCP (Model Context

Protocol), lo que facilita enormemente la conexión con herramientas y datos externos de forma segura.

Fuentes consultadas:

- <https://www.langchain.com/langgraph>
- <https://langchain-ai.github.io/langgraph>
- <https://github.com/langchain-ai/langgraph>
- <https://www.langchain.com/langgraph-platform-pricing>

TUTORIAL BÁSICO

Instalación

Para comenzar con LangGraph, es fundamental tener un entorno de Python actualizado. Se recomienda el uso de entornos virtuales o gestores modernos como uv.

- **Comando base:** `pip install -U langgraph` o `uv add langgraph`.
- **Dependencias necesarias:** Si planeas usar modelos de OpenAI o las integraciones estándar, instala también `langchain-openai` y `langchain`.
- **Checklist de configuración:**
 - Configura las variables de entorno para tus proveedores de LLM (ej. `OPENAI_API_KEY`).
 - **Recomendación:** Configura `LANGSMITH_TRACING=true` y tu `LANGSMITH_API_KEY`. Según mi experiencia, intentar depurar grafos complejos con ciclos sin trazabilidad visual es extremadamente ineficiente.

Uso en el día a día

LangGraph no es "LangChain 2.0", es un cambio de paradigma hacia **máquinas de estados**.

- **Define el Estado (State):** Utiliza `TypedDict` o `Pydantic` para definir qué información fluye por el grafo. Mi consejo es mantener el estado lo más limpio posible; no satures el estado con datos voluminosos como documentos completos si no son estrictamente necesarios para la lógica del siguiente nodo.
- **Nodos como funciones:** Cada nodo debe ser una función de Python pura que reciba el estado y devuelva una actualización. Al usarlo te das cuenta de que la clave es la **atomicidad**: un nodo, una tarea.
- **Compilación:** Una vez definidos nodos y bordes, usa `graph = builder.compile()`. Este objeto compilado es el que invocarás con `.invoke()`.

Trucos de experto

- **Reducers para Listas de Mensajes:** En el estado, etiqueta tus listas de mensajes con `Annotated[list, add_messages]`. Esto permite que LangGraph gestione automáticamente la concatenación de la historia de chat en lugar de sobrescribirla.
- **Human-in-the-loop:** Lo que más me gusta es la capacidad de usar breakpoints. Puedes configurar el grafo para que se detenga antes de ejecutar un nodo crítico (como una compra o un borrado de base de datos), permitiendo que un humano valide el estado actual antes de continuar.
- **Persistencia con Checkpointers:** Implementa `MemorySaver`. Esto permite que el agente "recuerde" hilos de conversación de forma nativa incluso si la aplicación se reinicia, almacenando el checkpoint del estado en cada paso.
- **Visualización en tiempo real:** Usa `display(Image(graph.get_graph().draw_mermaid_png()))` durante el desarrollo. En mi opinión profesional, ver el flujo visualmente ayuda a detectar bucles infinitos lógicos que el código a veces oculta.

Posibles problemas/incidencias

- **Bucles Infinitos:** Es común que un agente entre en un ciclo repetitivo si la lógica de salida del nodo no es clara. **Consejo:** Implementa siempre un contador de reintentos (`retry_count`) en tu estado y un borde condicional que fuerce el fin (END) tras N intentos.
- **InvalidUpdateError:** Ocurre cuando múltiples nodos intentan sobrescribir la misma clave del estado en un paso paralelo sin un reducer definido. Asegúrate de que cada clave tenga una regla de fusión clara.
- **Incompatibilidad de Esquemas:** Si usas `Pydantic` para el estado, recuerda que la validación ocurre al inicio del nodo. Errores de tipo durante la ejecución pueden ser difíciles de trazar si no se usa `LangSmith`.

Otros

- **LangGraph Studio:** Si buscas una experiencia más visual para prototipar, utiliza `LangGraph Studio` (disponible para macOS). Facilita la edición del estado "en vivo" para probar cómo reacciona el agente a diferentes inputs sin reiniciar el script.
- **Versatilidad de Modelos:** Aunque nace del ecosistema `LangChain`, puedes usar `LangGraph` con cualquier LLM (Anthropic, Google, locales con Ollama) siempre que la función del nodo devuelva el formato de diccionario esperado por el estado.

PREGUNTAS FRECUENTES

¿Qué es LangGraph y en qué se diferencia de LangChain?

LangGraph es un framework de orquestación de bajo nivel diseñado para construir agentes de IA con estados complejos y lógica cíclica. Mientras que LangChain estándar se enfoca en cadenas lineales de un solo paso, LangGraph permite crear grafos donde los procesos pueden retroceder, iterar y mantener un control preciso sobre el flujo de ejecución a través de nodos y aristas.

¿Es LangGraph una tecnología de código abierto?

Sí, la librería central de LangGraph es Open Source y se distribuye bajo la licencia MIT. Esto permite su uso, modificación y distribución tanto para fines personales como comerciales sin costes de licencia por el software base, evitando además el bloqueo con proveedores específicos.

¿Cuáles son los costes asociados al uso de LangGraph en un entorno profesional?

La librería es gratuita, pero los servicios de despliegue y observabilidad a través de LangGraph Cloud y LangSmith siguen un modelo freemium. Existe un plan gratuito para desarrolladores (hasta 5k trazas/mes), un plan Plus desde 39\$ por asiento mensual para despliegues gestionados, y planes Enterprise con precios personalizados para necesidades de alta escalabilidad o despliegues en nubes privadas (VPC).

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

LangGraph cumple con estándares internacionales como SOC2 y GDPR. Los datos enviados para monitorización a través de LangSmith no se utilizan para el entrenamiento de modelos de lenguaje externos. Al ser una librería instalable, el procesamiento de los datos ocurre en la infraestructura que el desarrollador elija (como contenedores Docker), manteniendo el control sobre la propiedad intelectual y el flujo de información.

¿Qué significa la funcionalidad 'Human-in-the-loop'?

Es una capacidad nativa que permite detener la ejecución del agente de forma programada para esperar la intervención humana. Un profesional puede revisar, editar o aprobar el estado del agente antes de que este continúe con el siguiente nodo del grafo, lo cual es crítico en sectores que requieren supervisión como el legal o el financiero.

¿Qué nivel de conocimientos técnicos se requiere para implementarlo?

El nivel técnico requerido es alto. No es una solución 'no-code'; requiere dominio de lenguajes de programación como Python o TypeScript, además de familiaridad con programación asíncrona, decoradores y tipos de datos (como Pydantic). Es una herramienta dirigida a ingenieros de software y arquitectos de IA.

¿Cómo gestiona LangGraph la persistencia y la memoria en tareas largas?

Incluye un sistema de 'Checkpointing' que guarda automáticamente el estado del grafo en cada paso. Esto permite que, si ocurre un error o una pausa, el agente pueda retomar la tarea exactamente donde se quedó. Además, soporta memoria multinivel, diferenciando entre la memoria de corta duración (dentro de una sesión) y la memoria de larga duración (entre distintas conversaciones).

¿Es compatible con otros modelos de IA y herramientas externas?

Sí, LangGraph se integra nativamente con el ecosistema LangChain, lo que le permite conectar con cientos de modelos (OpenAI, Anthropic, Ollama), bases de datos vectoriales y herramientas de terceros. También es compatible con el protocolo MCP (Model Context Protocol) para estandarizar la conexión entre modelos y herramientas externas.

¿Se puede utilizar LangGraph para aplicaciones en producción?

Está diseñado específicamente para llevar prototipos de agentes a producción de forma predecible. Empresas como Uber y J.P. Morgan han adoptado esta tecnología, lo que avala su capacidad para manejar flujos de trabajo de nivel empresarial que requieren alta fiabilidad y escalabilidad.

CONTRATOS Y CONDICIONES

Opinión inicial

Tras verificar los contratos y condiciones de LangChain Inc. (empresa matriz), LangGraph presenta un ecosistema híbrido desde el punto de vista legal. La librería core bajo licencia MIT ofrece la máxima libertad para empresas españolas, permitiendo el control total del flujo de datos ("data plane") si se auto-aloja. Sin embargo, el uso de la plataforma LangGraph Cloud y LangSmith para observabilidad introduce la legislación de EE.UU. (Delaware) como marco contractual principal. En mi opinión profesional, el mayor valor legal de LangGraph es la funcionalidad "Human-in-the-loop", que facilita el cumplimiento del futuro Reglamento de IA (AI Act) al permitir la supervisión humana efectiva en procesos automatizados de alto riesgo.

Principales recomendaciones

- **Privacidad desde el diseño:** Si se manejan datos de salud o financieros, optar por el despliegue "Self-Hosted" o "Hybrid" para garantizar que el procesamiento de datos personales no abandone la infraestructura de la empresa o territorio UE.
- **Configuración de Telemetría:** Al usar la CLI de LangGraph, desactivar activamente la telemetría mediante la variable de entorno LANGGRAPH_CLI_NO_ANALYTICS=1 para evitar el envío de metadatos de uso a servidores externos.
- **Anonimización en Origen:** Al integrar con LangSmith para depuración, implementar filtros de desidentificación (redacción de PII) antes de que las trazas salgan del entorno local hacia la nube de LangChain.

Ley de Inteligencia Artificial (AI Act)

- **Supervisión Humana:** La arquitectura de grafos de LangGraph es ideal para cumplir con el Art. 14 del AI Act, ya que permite definir "nodos de aprobación" obligatorios donde un humano debe validar la salida antes de avanzar.
- **Trazabilidad:** El sistema de "Checkpointing" y persistencia de estados facilita el cumplimiento de las obligaciones de registro de actividad (logging) requeridas para sistemas de IA de alto riesgo.

Privacidad y protección de datos

- **Responsabilidades:** Bajo el modelo de responsabilidad compartida, LangChain es el encargado del tratamiento (Processor) solo si se usa su nube; el usuario es siempre el Responsable del Tratamiento de los datos de entrada/salida.
- **Ubicación de los datos:** Los servicios Cloud se alojan por defecto en GCP (Google Cloud) con opciones de región en EE.UU. Aunque disponen de opción EU en planes Enterprise, por defecto puede haber transferencias internacionales.
- **Transferencia internacional:** LangChain cuenta con un Data Processing Addendum (DPA) que incorpora Cláusulas Contractuales Tipo (SCCs), necesario para legitimar el flujo de datos desde España a EE.UU. conforme al RGPD.
- **Derechos ARCO:** La persistencia de hilos ("threads") en bases de datos PostgreSQL (en langgraph up) requiere que la empresa implemente mecanismos para localizar y borrar datos personales de usuarios específicos si estos ejercen su derecho de supresión.

Propiedad intelectual

- **Propiedad de datos:** Los términos de servicio especifican claramente que el "Customer Data" (entradas y salidas) pertenece íntegramente al usuario.
- **Propiedad del resultado:** El código del grafo desarrollado por la empresa sobre la librería MIT es propiedad intelectual de la empresa. LangChain no reclama derechos sobre la lógica de negocio construida sobre el framework.

Usos y prohibiciones

- **Usos prohibidos:** Los términos prohíben explícitamente el uso de LangGraph para "Actividades de Alto Riesgo" que puedan causar daños físicos o muerte, así como realizar ingeniería inversa para crear productos competidores.
- **Usos admitidos:** Desarrollo de aplicaciones comerciales, agentes autónomos y sistemas de orquestación de LLM bajo licencia permisiva.

Seguridad y certificaciones

- **Seguridad:** Utiliza cifrado AES-256 para datos en reposo y TLS 1.2+ para datos en tránsito. Los usuarios pueden añadir cifrado propio para los checkpoints usando la clave LANGGRAPH_AES_KEY.

- **Certificaciones:** LangChain mantiene certificaciones SOC 2 Tipo II, y declara cumplimiento con GDPR e HIPAA para sus servicios gestionados.

Otros

- **Uso de Datos para Entrenamiento:** Según sus condiciones actuales, LangChain declara explícitamente que no utiliza los datos enviados a LangSmith o LangGraph Cloud para entrenar sus modelos ni los de terceros.

Fuentes consultadas:

- [Condiciones de Servicio de LangChain \(LangSmith/LangGraph\)](#)
- [Política de Privacidad y RGPD](#)
- [Documentación de Privacidad y Almacenamiento de LangGraph](#)
- [Modelo de Responsabilidad Compartida y Seguridad](#)
- [Licencia MIT de LangGraph en GitHub](#)
- [Precios y Planes de Hosting \(Cloud vs Hybrid\)](#)

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.