

Tu primer agente con Claude Code

Bucle de percepción, decisión, acción con herramientas reales.

CONCEPTOS CLAVE

- Un **agente** cierra el bucle *objetivo* → *razonar* → *actuar* → *observar*. Diferencia clave con chatbot: *actúa*.
- Pipeline cloud (Claude Code) y pipeline libre (opencode + Ollama) son equivalentes en capacidad.
- CLAUDE.md = system prompt jerárquico: global, proyecto, subagentes, hooks.
- Herramientas nativas: Read, Edit, Bash, Grep, WebSearch, Agent, TaskCreate.

EJEMPLO A MANO

Sesión real, 7 turnos: refactor con retry sobre 3 archivos.

Quién	Qué hace
Usuario	Pide añadir retry a funciones que llaman a Anthropic API.
Agente → Grep	Busca <code>anthropic.messages.create</code> en <code>src/</code> .
Resultado	3 archivos encontrados.
Agente → Read	Lee los 3 archivos.
Agente → Edit	Inserta <code>@retry(stop_after_attempt(3))</code> .
Agente → Bash	<code>pytest -k retry verde</code> .
Agente	Reporta: 3 funciones modificadas, 1 test añadido.

CONCEPTO → APLICACIÓN: TAREAS PARA UN AGENTE HOY

Tarea	Caso	KPI
Refactor automatizado	Migrar 200 ficheros Py2 a Py3 (BB-VA).	Tests OK / día.
Documentación	Generar README de módulo nuevo.	Tiempo a entrega.
Revisión de PR	Revisar PR antes de merge.	Issues detectados.
Análisis exploratorio	Cargar CSV y producir informe.	Tiempo a insight.
Triage de tickets	Clasificar y responder Zendesk.	% auto-resueltos.

Idea clave del Día 13. Un agente es un junior nuevo: necesita instrucciones claras, revisión y barandillas. La calidad de tu CLAUDE.md (rol + convenciones + restricciones + contexto + política IA) determina la calidad del agente.

Para llevarte: lectura Anthropic, *Claude Code overview*; práctica: escribir un CLAUDE.md para un repo propio y pedir al agente que produzca su README.