

Día 13 · Tema 5 — Agentes: del LLM al bucle con herramientas

1. Tres niveles de automatización

- **Script:** $a = f(x)$. Función determinista; no ve resultados ni reescribe planes.
- **Chatbot:** $a_t = \pi(x_{1:t})$. Responde turno a turno pero no actúa sobre el entorno.
- **Agente:** $a_t = \pi(c_t)$, donde c_t acumula resultados de herramientas, lecturas y escrituras (ver el bucle del agente del Día 01).

2. El bucle del agente con herramientas reales

Formalizamos el agente como un proceso de decisión con espacio de acciones $\mathcal{A} = \mathcal{A}_{\text{texto}} \cup \mathcal{A}_{\text{tool}}$:

$$\pi_\theta : c_t \mapsto (a_t, c_{t+1}), \quad c_{t+1} = c_t \cup \{a_t, \text{exec}(a_t)\}. \quad (1)$$

c_t contexto en el paso t : prompts, respuestas previas, traza de tools
 a_t acción del agente; o texto (mensaje al usuario) o llamada a tool
 exec entorno: ejecuta el tool y devuelve el resultado
 π_θ política, encarnada en el LLM

La traza es *el* artefacto auditable del agente: cada a_t y cada $\text{exec}(a_t)$ queda registrado. Si algo sale mal, se reproduce.

3. Llamada a herramientas en Claude Code

Una herramienta se define con una firma JSON Schema. El modelo decide cuándo llamarla y con qué argumentos:

$$a_t = \begin{cases} \text{Read}(\text{path}) & \text{leer un fichero} \\ \text{Edit}(\text{path}, \text{old}, \text{new}) & \text{editar contenido} \\ \text{Bash}(\text{cmd}, \text{timeout}) & \text{ejecutar comando} \\ \text{WebSearch}(\text{q}) & \text{consultar web} \\ \dots & \end{cases} \quad (2)$$

La respuesta $\text{exec}(a_t)$ se concatena al contexto como un mensaje de rol *tool* y el modelo decide el siguiente paso. **No es recursión profunda: es una iteración acotada con criterio de parada explícito (*tarea cumplida o abandono*).**

4. CLAUDE.md y la memoria persistente

Las preferencias del usuario, las convenciones del repositorio y las restricciones del proyecto viven en un `CLAUDE.md` en la raíz. El agente lo lee al inicio de cada sesión y lo trata como *system prompt extendido*:

$$c_0 = \text{system prompt} \oplus \text{CLAUDE.md} \oplus \text{request del usuario}. \quad (3)$$

Si el usuario dice “de ahora en adelante usa siempre tabs en lugar de espacios”, el agente lo escribe en `CLAUDE.md`; en la siguiente sesión esa preferencia ya está cargada en c_0 .

5. Coste por sesión

Una sesión de agente típica (10–30 minutos) intercambia varias docenas de mensajes con el modelo. El consumo se aproxima por

$$N_{\text{tokens}}^{\text{sesión}} \approx N_{\text{turns}} (\bar{n}_{\text{ctx}} + \bar{n}_{\text{resp}}), \quad (4)$$

N_{turns} número de turnos (LLM call + tool call counted)
 $\bar{n}_{\text{ctx}}$ tokens promedio del contexto en cada llamada
 $\bar{n}_{\text{resp}}$ tokens promedio de la respuesta

Para $N_{\text{turns}} = 40$, $\bar{n}_{\text{ctx}} = 6000$, $\bar{n}_{\text{resp}} = 400$ y Sonnet 4.5 (3/15 \$/M), $C \approx 40 \cdot (6000 \cdot 3 + 400 \cdot 15) / 10^6 = 40 \cdot 0,024 \approx 0,96 \$$.
Una sesión productiva $\approx 1 \$$.

6. Patrones que funcionan

1. **Definir tarea + criterio de fin.** “Implementa X. Termina cuando los tests pasen.”

2. **Romper en subtareas explícitas.** El agente lleva un TODO list; lo va tachando.
3. **Verificar siempre.** Cada cambio del agente debe ser ejecutado/testado por él mismo antes de declararlo terminado.
4. **Diff antes de commit.** El humano revisa el delta, no la descripción.

IDEA CLAVE. Lo que en 2010 era *escribir código* es ahora *describir intención, supervisar la traza y validar resultados*. El programador no ha desaparecido: ha subido un nivel de abstracción.

7. Cuándo NO usar un agente

Tres casos donde el agente es peor que un script: **(a)** si la tarea está claramente formalizada (parsear un CSV con un schema fijo) — un script es más rápido, barato y reproducible; **(b)** si el coste de un error es alto y el agente no tiene rollback (borrar la base de datos de producción); **(c)** si la latencia importa más que la inteligencia (responder en < 50 ms a un click web) — un agente que llama al LLM en cada interacción no escala.