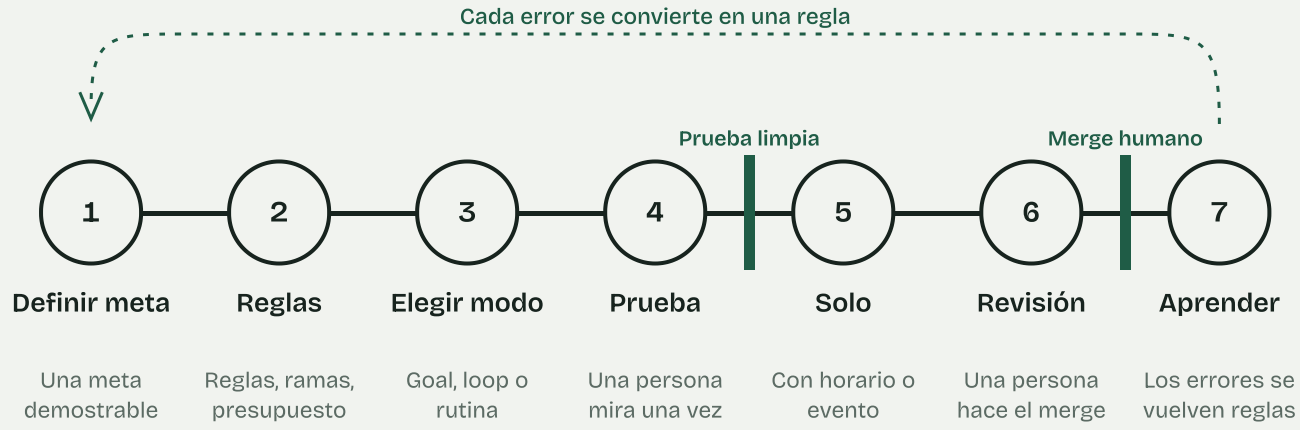


Dominik Heinrich, arquitecto de soluciones DevOps multicloud e IA

Agentes que trabajan mientras duermes: goals, loops y rutinas

Cómo llevo a los equipos del ping-pong de prompts a agentes que funcionan solos, con una meta clara, un horario y una revisión humana. Sirve para Claude Code y Codex.



Siete pasos, dos controles. Los agentes trabajan solos; las personas deciden qué se publica.

De plataformas en producción que opero así, anonimizado

5.000+	Cada hora	Cada noche	Cada día
pull requests fusionados en dos plataformas	comprobación de los flujos de reserva, sin supervisión	prueba de restauración de las copias	informe de costes en el chat del equipo

Los siete pasos

Del ping-pong de prompts a agentes que trabajan con horario. Márcalos cuando estén en marcha.

01 Define qué es terminado antes de empezar

- [ ] Escribe la meta como algo que el agente pueda demostrar: 'npm test termina con 0', no 'mejóralo'.
- [ ] **Delimita el alcance:** qué carpetas, qué tickets y qué está prohibido.
- [ ] Mete un límite en el propio objetivo, por ejemplo 'o para tras 20 turnos e informa de lo que falta'.

02 Pon las reglas

- [ ] Un archivo de instrucciones por repositorio con las reglas que el agente debe cumplir.
- [ ] **Solo ramas:** el agente abre pull requests, nunca hace merge ni despliega en producción.
- [ ] Ninguna credencial de producción, solo cuentas de prueba, y un presupuesto de tokens por ejecución.

03 Elige el modo adecuado

- [ ] **Meta clara:** un goal (/goal en Claude Code, goal mode en Codex).
- [ ] Comprobación periódica mientras trabajas: un loop con intervalo, por ejemplo /loop 15m.
- [ ] Tiene que funcionar con el portátil cerrado: una rutina en la nube, lanzada por horario, por una llamada a la API o por un evento de GitHub.

04 Prueba con una persona mirando

- [ ] Ejecútalo una vez de forma interactiva y lee cada paso.
- [ ] Corrige las instrucciones donde el agente se equivocó, no solo el resultado.
- [ ] Solo funciona sin supervisión cuando una prueba sale limpia.

05 Déjalo funcionar solo

- [ ] **Empieza en pequeño:** un repositorio, un tipo de tarea, como actualizar dependencias, arreglar tests fallidos o un informe nocturno.
- [ ] Cada ejecución escribe un resumen corto: qué hizo, qué dejó y qué necesita a una persona.
- [ ] Una alerta llega al chat del equipo cuando una ejecución falla o tarda mucho más de lo normal.

06 Revisión antes de publicar nada

- [ ] Una persona revisa y hace el merge, igual que con los pull requests de personas.
- [ ] **Reglas fijas:** login, pagos, borrado de datos y permisos siempre necesitan a una persona.
- [ ] Primero staging, luego producción, con smoke tests después de cada despliegue.

07 Aprende y amplía

- [ ] Cada error se convierte en una regla del archivo de instrucciones, para que la siguiente ejecución lo evite.
- [ ] Mide las horas ahorradas, el coste por ejecución y el porcentaje de ejecuciones que necesitaron ayuda.
- [ ] Añade el siguiente tipo de tarea solo cuando el actual lleve dos semanas funcionando sin sobresaltos.

Cómo implantarlo según el tamaño de la empresa

Startup: una rutina para el trabajo pesado, como actualizar dependencias, tests fallidos y el informe semanal. Suele estar en marcha en una semana.

Empresa mediana: un equipo piloto con archivos de instrucciones compartidos y un presupuesto de tokens, luego el resto de equipos.

Gran empresa: ajustes gestionados que los usuarios no pueden cambiar, una lista aprobada de modelos y herramientas, registros de auditoría y modelos locales para el código sensible.

Los errores que más veo

- Un objetivo como 'mejora el código' que nunca termina y quema tokens.
- Dejar que el agente dé su trabajo por terminado sin tests que lo demuestren.
- Dejarlo solo antes de una prueba supervisada.
- Agentes con permisos de merge o credenciales de producción.
- Corregir el resultado a mano en vez de las instrucciones, y mañana vuelve el mismo error.

Cómo podemos trabajar juntos

La misma calidad para una startup de diez personas y para un grupo global. El formato se adapta a ti.

Startups

Responsable de DevOps e IA a tiempo parcial

Unos días al mes: arquitectura, pipeline de releases y funciones de IA, para que los fundadores se centren en los clientes. Del prototipo a producción sin reescribir.

Pymes

Un proyecto con un resultado claro

Un objetivo definido, como una factura cloud más baja, una migración o los primeros agentes de IA en producción, con alcance cerrado, avances semanales y una entrega que tu equipo pueda operar.

Grandes empresas

Un arquitecto dentro de tus equipos

Integrado con tus equipos de plataforma, seguridad y producto: decisiones de arquitectura, gobernanza de la IA y entornos regulados. Para alcances mayores traigo un equipo senior de confianza.

Cómo trabajo

Decisiones por escrito: registros de decisiones de arquitectura con sus pros y contras, para que tu equipo sepa el porqué, no solo el qué.

Operado, no solo entregado: SLOs, alertas sobre resultados de negocio y una rutina de incidencias.

El conocimiento se queda contigo: runbooks, documentación y talleres para tu equipo.

Seguro por defecto: OIDC e identidades gestionadas, mínimo privilegio, secretos en un almacén seguro y trazabilidad.

Medido: coste, tiempo de entrega, disponibilidad y horas ahorradas, con un informe semanal.

Un único interlocutor que habla con ingeniería y con dirección, en alemán, inglés y español.



¿Quieres que los agentes de tu equipo trabajen solos y de forma segura?

Reservar una llamada de 30 minutos

E-mail mail@dominikheinrich.ch
Web dominikheinrich.ch

Disponible en todo el mundo en remoto. Presencial bajo petición, según el país.