

Del diseño al despliegue, a un clic por paso.

El ciclo completo en 6 fases. En cada una, acciones para copiar en un clic: **sh** = comando de terminal (instalar, configurar, ejecutar); **IA** = prompt para tu editor agéntico — **cópialo y pégalo en Claude Code / Cursor sobre tu proyecto** (o «abrir chat» para la versión web). Escribe tu **módulo o tarea** en el campo de contexto y se sustituye en los <...> al copiar; el chip **tpl** marca lo que debes editar antes de ejecutar y **riesgo** marca comandos que requieren revisión humana. El **nivel** ajusta el rigor y el **stack** cambia entre Node/Python. La Suite no reemplaza tus herramientas estándar: las **complementa** con contexto trazable, control y evidencia.

01 Diseño

C4 model ↗

Define la arquitectura mínima viable y registra cada decisión.

N1 **IA** Propón la arquitectura mínima para mi proyecto: componentes, límites y un diagrama textual. Justifica cada decisión en una línea.

N1 **sh** `mkdir -p docs/adr && printf '# ADR-0001: decisión inicial\n' > docs/adr/0001-inicio.md`

N2 **IA** **tpl** Escribe un ADR para <decisión>: contexto, opciones consideradas, decisión y consecuencias.

N2 **sh** `npx -y adr-log -o docs/adr/README.md`

N3 **IA** Modela el sistema con el C4 model (niveles 1-2) y prioriza los atributos de calidad ISO/IEC 25010 relevantes, con justificación.

+ complementa lo estándar · **SP-LLMGraph** prepara el contexto real del proyecto; **SP-DevControl** registra las decisiones (ADR) bajo aprobación.

Desglosa el trabajo y fija convenciones de versionado y commits.

N1 **sh** `git init && git branch -M main && npm init -y`

N1 **IA** **tpl** Desglosa **«objetivo»** en tareas pequeñas con criterios de aceptación y ordénalas por dependencia.

N2 **sh** `npx -y commitizen init cz-conventional-changelog --save-dev --save-exact`

N2 **sh** **tpl** **riesgo** `gh issue create --title '<tarea>' --body '<criterios de aceptación>'`

N3 **IA** Define las métricas DORA objetivo para este proyecto y qué instrumentaremos para medirlas.

+ **complementa lo estándar** · **SP-DevSteps** ordena el proceso por pasos; **SP-DevControl** fija y valida las convenciones desde el inicio.

Implementa con estilo consistente y cambios revisables.

N1 **sh** **riesgo** npm i -D eslint prettier && npx eslint --init

N1 **sh** printf 'root = true\n\n[*]\nindent_style = space\nindent_size = 2\n' > .editorconfig

N1 **IA** **tpl** Implementa **funcionalidad**. Muéstrame el diff antes de aplicarlo y sigue las convenciones del repo.

N2 **sh** **riesgo** pip install pre-commit && pre-commit install

N2 **IA** **tpl** Prepara el contexto con SP-LLMGraph y refactoriza **módulo** sin cambiar el comportamiento; añade pruebas.

N3 **IA** **tpl** Introduce feature flags en **cambio** para permitir despliegue progresivo y rollback rápido.

+ complementa lo estándar · **SP-LLMGraph** reduce tokens al seleccionar contexto relevante; **SP-DevControl** exige aprobación del diff; **SP-ForgePilot** añade skills de terceros con firma.

Prueba con criterio siguiendo la pirámide; mide cobertura.

N1 **sh** **riesgo** `npm i -D jest`

N1 **sh** `npm test`

N2 **sh** `npx jest --coverage`

N2 **IA** **tpl** Genera pruebas unitarias para **«módulo»** cubriendo casos límite; deja la cobertura $\geq 80\%$.

N3 **sh** **riesgo** `npm init playwright@latest && npx playwright test`

N3 **IA** **tpl** Diseña un plan de pruebas de carga para **«endpoint»** con umbrales de latencia y tasa de error.

+ complementa lo estándar · **SP-ForgePilot** instala skills de testing; **SP-DevControl** revisa el cambio antes de aplicarlo.

Verifica seguridad, dependencias y cadena de suministro.

N1	sh	riesgo	<code>npm audit --audit-level=high</code>
N1	sh	riesgo	<code>docker run --rm -v "\$PWD:/repo" zricethezav/gitleaks:latest detect --source=/repo -v</code>
N2	sh	riesgo	<code>docker run --rm -v "\$PWD:/src" aquasec/trivy fs /src</code>
N2	IA		Revisa este diff contra OWASP Top 10 y enumera los riesgos con severidad y remediación.
N3	sh	tpl	<code>gh attestation verify <artefacto> --owner PROjas01</code>
N3	sh		<code>conftest test deploy/ --policy policy/</code>
N3	IA	tpl	Verifica el cumplimiento de OWASP ASVS nivel 2 para <componente> y lista las brechas.

+ complementa lo estándar · **SP-DevControl** aplica política de comandos y deja auditoría;
SP-LinkAnalyzer guarda evidencia con hash y fecha.

06 Despliegue

SLSA v1.2 ↗

Entrega continua, firmada y bajo aprobación humana.

N1 **IA** Genera un workflow de GitHub Actions que corra lint + test + build en cada push a main.

N1 **sh** `mkdir -p .github/workflows`

N2 **sh** **riesgo** `docker build -t app:$(git rev-parse --short HEAD) .`

N2 **sh** `terraform init && terraform plan`

N3 **sh** **riesgo** `terraform apply` # solo tras aprobación humana en SP-DevControl

N3 **IA** Propón una estrategia de despliegue canary con criterios de rollback basados en SLO.

+ complementa lo estándar · **SP-DevControl** aprueba el despliegue (fail-closed) y lo audita; **SP-DevSteps** ordena el pipeline.

◆ Adáptalo a tu vertical

Mismo ciclo; cambian las herramientas de la Suite y las fuentes de terceros a alimentar en ForgePilot.

DESARROLLO

Software

COMPLEMENTO · SUITE

- SP-DevControl
- SP-DevSteps
- SP-LLMGraph
- SP-ForgePilot

ESTÁNDAR / TERCEROS

- awesome-claude-code-subagents
- wshobson/agents
- OpenAI Cookbook

ANÁLISIS

Datos / IA

COMPLEMENTO · SUITE

- SP-LLMGraph
- SP-DevControl
- SP-LocalTokens

ESTÁNDAR / TERCEROS

- VoltAgent · data-ai
- postgres-mcp

AUTOMATIZACIÓN Industrial / IoT COMPLEMENTO · SUITE - SP-ForgePilot - SP-DevControl - SP-LLMGraph ESTÁNDAR / TERCEROS - ModbusMCP - opcua-mcp	CONTENIDO Marketing COMPLEMENTO · SUITE - SP-LinkAnalyzer - SP-LLMGraph - SP-DevControl ESTÁNDAR / TERCEROS - anthropics/skills - awesome-ai-marketing
CRM / ATENCIÓN Ventas COMPLEMENTO · SUITE - SP-Asesor1Pro - SP-LLMGraph - SP-LocalTokens ESTÁNDAR / TERCEROS - business-prompts - copilot-prompts	FINANZAS / ERP Operaciones COMPLEMENTO · SUITE - SP-DevControl - SP-LocalTokens - SP-ForgePilot ESTÁNDAR / TERCEROS - postgres-mcp - copilot-prompts