

Día 2: Harness avanzado y SDD

Se amplía el arnés del agente con tres piezas: comandos personalizados (prompts reutilizables con /nombre), skills (instrucciones en SKILL.md que el agente carga cuando las necesita, propias o de terceros) y MCP (protocolo para conectar herramientas externas como Chrome DevTools, Context7, GitHub, Figma o Supabase). Después se presenta el Spec-Driven Development (SDD): definir primero qué debe hacer el sistema y usar esa especificación para guiar y validar la implementación. Existen tres enfoques (spec-first, spec-anchored, spec-as-source), siendo spec-anchored el recomendado para empezar. Los requisitos se escriben en notación EARS para evitar ambigüedades. El flujo tiene siete pasos: constitución, especificación, clarificación, plan, tareas, implementación y validación, con bucle de mantenimiento. Todo se automatiza con la skill /sdd y nueve comandos (/sdd-spec, /sdd-plan, /sdd-implement, /sdd-validate, etc.). En el menú de la izquierda podrás encontrar el documento correspondiente al curso, así como el contenido en video

INDICE

- 3. 03 – Harness engineering avanzado
 - Comandos personalizados
 - Agent Skills (propias y de terceros, starter pack)
 - MCP (ventajas, creación, directorios, MCPs esenciales)
 - Uso de Chrome DevTools para testing
- 3. 04 – Spec-Driven Development (SDD)
 - Qué es y flujo clásico vs. flujo SDD
 - Taxonomía SDD (spec-first, spec-anchored, spec-as-source)
- EARS
- Estructura de proyecto SDD
- SDD paso a paso (constitución, spec, clarificación, plan, tareas, implementación, validación)
- Automatización de SDD con skills y comandos