

Spec Kit · escribí los specs antes que Claude escriba código

CATEGORÍA · CÓDIGO Y DESARROLLO

Toolkit MIT oficial del equipo de GitHub (~96k stars) que mete Spec-Driven Development en Claude Code. Instalás specify-cli con uv, corrés specify init mi-proyecto --integration claude y dentro de Claude Code aparecen 9 slash commands: constitution → specify → clarify → checklist → plan → tasks → ta

QUÉ VAS A ENCONTRAR

- 01 Qué es y qué resuelve
- 02 Para qué sirve
- 03 Dónde aplicarlo
- 04 Instalación en 3 rutas (con comandos)
- 05 El código en GitHub
- 06 Comprobación y errores frecuentes

FICHA DEL RECURSO

Qué es, para qué sirve y cómo se implementa

FUNCIÓN

Toolkit MIT oficial del equipo de GitHub (~96k stars) que mete Spec-Driven Development en Claude Code. Instalás specify-cli con uv, corrés specify init mi-proyecto --integration claude y dentro de Claude Code aparecen 9 slash commands: constitution → specify → clarify → checklist → plan → tasks → ta

PARA QUÉ SIRVE

Toolkit MIT oficial del equipo de GitHub (~96k stars) que mete Spec-Driven Development en Claude Code. Instalás specify-cli con uv, corrés specify init mi-proyecto --integration claude y dentro de Claude Code aparecen 9 slash commands: constitution → specify → clarify → checklist → plan → tasks → taskstoissues → analyze → implement. Vos dictás qué construir, quién va a usarlo y cuál es el resultado final; Claude pregunta lo que no quedó claro, te arma el plan técnico, lo rompe en tareas chicas y va tachando cada una. Funciona con 30+ agentes (Cursor, Copilot, Gemini, Codex, Windsurf, Goose).

DÓNDE APLICARLO

Nivel Intermedio. Temas: spec-kit, specify. Incluye el código de github/spec-kit.

CÓMO IMPLEMENTARLO

- 1 Ruta rápida: pégle a Claude Code el enlace del repositorio con el prompt de la página siguiente — instala y comprueba solo.
- 2 Ruta manual: clona el repositorio y corre los comandos de instalación de tu lenguaje.
- 3 Revisa el README antes de instalar: requisitos, versión y licencia.
- 4 Pruébalo en un caso pequeño antes de usarlo en trabajo real.
- 5 Revisa qué permisos y llaves pide antes de conectarlo a tus cuentas.

EL RECURSO EN DETALLE

Qué resuelve y cómo funciona

Toolkit MIT oficial del equipo de GitHub (~96k stars) que mete Spec-Driven Development en Claude Code. Instalás `specify-cli` con `uv`, corrés `specify init mi-proyecto --integration claude` y dentro de Claude Code aparecen 9 slash commands: `constitution` → `specify` → `clarify` → `checklist` → `plan` → `tasks` → `taskstoissues` → `analyze` → `implement`.

Vos dictás qué construir, quién va a usarlo y cuál es el resultado final; Claude pregunta lo que no quedó claro, te arma el plan técnico, lo rompe en tareas chicas y va tachando cada una. Funciona con 30+ agentes (Cursor, Copilot, Gemini, Codex, Windsurf, Goose).

La cubre los pre-requisitos honestos (Python 3.11+, `uv`, `git`), los 2 caminos de instalación (`uv tool install` manual o pegale el prompt a Claude), los 9 comandos en detalle con qué produce cada uno, un flujo end-to-end completo con una to-do app (Vos escribís → Claude responde → Archivo que aparece), 4 prompts copy-paste para las fases que importan (`constitution`, `specify`, `plan`, `implement`), 5 trampas honestas (incluye comparación honesta vs `/plan` mode nativo de Claude Code), y cross-links a `aprende-skill`, `crea-claude-skills`, `claude-codex-equipo`, `dominar-cowork`, `cual-plan-claude` e `instalar-claude-code`.

INSTALACIÓN

spec-kit

Toolkit que mete Spec-Driven Development en Claude Code: dictás los specs, él arma el plan y va tachando tareas.

EL CÓDIGO ESTÁ AQUÍ

<https://github.com/github/spec-kit>

Licencia MIT · Python

Ruta 1 · La más simple. Abre Claude Code, pégale este mensaje y él hace el resto:

PÉGASELO A CLAUDE CODE

Instala esto por mi y explicame lo que haces:

<https://github.com/github/spec-kit>

Antes de correr nada, leete el README y dime:

1. que hace exactamente y para que sirve
2. que requisitos necesito instalados
3. que permisos o llaves va a pedir

Luego instalalo paso a paso, avisame de cada comando antes de ejecutarlo, y al final comprueba que quedo funcionando.

Ruta 2 · Manual, si prefieres controlar cada paso desde la terminal:

INSTALACIÓN MANUAL · TERMINAL

```
# 1. Clona el repositorio
git clone https://github.com/github/spec-kit.git
cd spec-kit

# 2. Instala las dependencias (Python)
python3 -m venv .venv && source .venv/bin/activate
pip install -r requirements.txt

# 3. Sigue el ejemplo de uso del README para tu caso
```

Ruta 3 · Sin instalar nada. Entra al enlace de arriba y lee el README y los ejemplos para decidir si te sirve antes de bajarlo.

ANTES DE DARLO POR HECHO

Comprobación y errores frecuentes

- 1 No quedó instalado si no lo has probado: corre el ejemplo del README y comprueba que devuelve lo esperado.
- 2 Si un comando falla, casi siempre falta un requisito previo (versión de Node, Python o git). Pégale el error a Claude Code y pídele que lo resuelva.
- 3 Si pide una llave o token, créala en la cuenta correspondiente y guárdala fuera del código, en una variable de entorno.
- 4 Antes de conectarlo a cuentas reales, pruébalo con datos de prueba o una cuenta secundaria.
- 5 Comprueba la licencia si lo vas a usar en trabajo de cliente: define qué puedes hacer con el código.

DATOS DEL RECURSO

Tipo: Guía paso a paso

Categoría: Código y desarrollo

Nivel: Intermedio

Licencia: MIT

Lenguaje: Python

Temas: spec-kit, specify

Código: <https://github.com/github/spec-kit>

El ecosistema de IA cambia rápido: confirma en el repositorio que el proyecto sigue activo y que los comandos no han cambiado desde la fecha de esta ficha.

SIGUIENTE PASO

Instálalo hoy, en pequeño.

Elige la ruta 1, pégale el enlace a Claude Code y pruébalo en un caso real chico. Un recurso instalado y probado vale más que diez guardados para después.

MÁS GUÍAS GRATIS EN [377CREATIVE.COM/RECURSOS](https://377creative.com/recursos)