Claude Code con criterio operativo

Guía práctica para usar Claude Code con criterio operativo: cómo instalarlo, cómo trabajar con Git, cómo dividir tareas, cómo revisar cambios y cómo evitar que el agente rompa más de la cuenta.

1. Qué es Claude Code

Claude Code es un agente de programación que trabaja desde la terminal. Puede leer archivos, editar varios ficheros, ejecutar comandos, revisar errores, preparar cambios y ayudar a mantener una tarea de programación sin perder el hilo.

No es un chat bonito para pedir fragmentos de código. Es una herramienta para trabajar sobre un proyecto real, con permisos, cambios, pruebas y revisión.

Claude Code sirve para:

- entender un proyecto nuevo
- buscar dónde está un error
- editar varios ficheros a la vez
- ejecutar pruebas
- preparar cambios para revisar
- trabajar con Git
- dividir una tarea en pasos
- revisar antes de aceptar cambios
- mantener contexto en una tarea larga
- ayudar a documentar lo que se ha hecho

Claude Code no sustituye criterio. Sustituye tecleo, búsqueda, cambios repetidos, revisiones iniciales y parte del trabajo de ejecución.

2. La mentalidad correcta

Claude Code funciona bien cuando se le trata como un agente que puede cometer errores, no como una persona que siempre sabe lo que hace.

Por eso conviene:

- no dejarle cambiar todo el proyecto de golpe
- no aceptar cambios sin revisar
- no dejarle ejecutar cosas peligrosas sin permiso
- no dejarle tocar producción
- no dejarle inventar pruebas
- no dejarle cerrar una tarea sin decir qué ha hecho

3. Chat, Cowork y Code

Claude ofrece varias formas de usar su ecosistema. Para un trabajo de programación, lo importante es Code.

	Chat	Cowork	Code
Para	Preguntas, ideas, textos, dudas	Tareas de oficina, ficheros, navegador	Trabajar con código, Git, ficheros, pruebas
Ve ficheros	No	Sí	Sí
Puede editar ficheros	No	Sí	Sí
Puede ejecutar comandos	No	Limitado	Sí, con permisos
Usa Git	No	No	Sí
Puede revisar cambios	No	Sí	Sí
Puede trabajar en varias tareas	No	Sí	Sí
Puede tocar producción	No	Sí	Sí, si se le permite

Para programar, el centro es Code.

4. Desktop o terminal: cuál escoger

Claude Code puede usarse desde la app de escritorio o desde la terminal.

Desktop	Terminal
Más cómodo para empezar	Más rápido para quien ya trabaja en terminal
Más visual para revisar cambios	Más cerca del proyecto
Más fácil para ver ficheros	Más fácil de combinar con Git, tests y scripts
Más fácil para permisos	Más fácil de automatizar
Más fácil para sesiones largas	Más fácil de usar en servidor

Regla: empieza por Desktop si quieres ver cambios con calma. Pasa a terminal cuando ya sepas lo que haces y quieras más control.

5. Instalar Claude Code en Mac y Windows

6. Crear la app de escritorio

7. Diario desde la terminal

8. Modelo, esfuerzo y permisos

9. Tu workspace: el second brain

Estructura recomendada

```
workspace/
├ inbox/
├ tareas/
├ notas/
├ decisiones/
├ proyectos/
└ revisiones/
```

Qué es cada carpeta

- **inbox/**: ideas sueltas, notas rápidas, cosas que todavía no están claras
- **tareas/**: tareas en curso, pendientes, bloqueos
- **notas/**: notas de trabajo, resúmenes, aprendizajes
- **decisiones/**: decisiones tomadas y por qué se tomaron
- **proyectos/**: proyectos activos, estado, siguientes pasos
- **revisiones/**: revisiones hechas, errores encontrados, cambios revisados

Regla

Si Claude Code va a tocar varios ficheros, que deje rastro de lo que ha hecho.

10. Primer proyecto real

Elige una tarea pequeña, clara y fácil de revisar. Mejor empezar con cosas que puedan salir bien.

Buena tarea

- corregir un error
- añadir una prueba
- mejorar un mensaje de error
- documentar un comando
- revisar un fichero
- preparar un cambio pequeño

Mala tarea

- rehacer todo el proyecto
- cambiar varias cosas a la vez
- tocar producción
- dejar que invente pruebas
- aceptar cambios sin revisar

Regla 1: instalar de forma limpia

```
node -v
npm install -g @anthropic-ai/claude-code
cd ~/proyectos/mi-app
claude
```

Regla 2: revisar antes de aceptar

```
git status
git diff
```

Regla 3: no dejar que toque todo

Si Claude Code empieza a tocar muchos ficheros, para y pide que lo haga por partes.

Regla 4: probar después de cambiar

Si cambia algo, que ejecute pruebas o que te diga por qué no puede hacerlo.

Regla 5: no aceptar cosas que no entiendas

Si no entiendes el cambio, no lo apruebes todavía.

Regla 6: dejar rastro

Si Claude Code va a tocar varios ficheros, que deje rastro de lo que ha hecho.

11. Skills

Las skills son instrucciones que Claude Code puede usar para trabajar de forma más ordenada.

Sirven para que no tengas que repetir siempre el mismo tipo de petición.

Ejemplo

```
/skill revisar-cambios
/skill preparar-commit
/skill crear-tarea
/skill actualizar-diario
```

Qué hacen

- **revisar-cambios**: revisa cambios antes de aceptarlos
- **preparar-commit**: prepara un commit con criterio
- **crear-tarea**: crea una tarea con estructura clara
- **actualizar-diario**: actualiza el diario de trabajo

Regla

Si una petición se repite mucho, puede ser una skill.

12. Rutinas, automatizaciones y agentes

Claude Code puede ayudar a preparar rutinas, automatizaciones y agentes.

Ejemplos

- revisar cambios antes de aceptarlos
- preparar un commit
- crear una tarea
- actualizar el diario
- revisar errores
- preparar un informe
- ejecutar pruebas
- revisar ficheros

Regla

Si una petición se repite mucho, puede ser una skill.

13. Checklist y plan de 7 días

Checklist antes de trabajar

- Node.js instalado
- Claude Code instalado
- proyecto en carpeta clara
- Git activo
- permisos revisados
- tarea pequeña
- diario activado
- revisión activa

Plan de 7 días

- Día 1: instalar Claude Code y probar ficheros
- Día 2: probar Git y revisar cambios
- Día 3: probar tareas pequeñas
- Día 4: probar permisos y errores
- Día 5: probar skills y rutinas
- Día 6: probar diario y revisiones
- Día 7: revisar qué funciona y qué no

Cierre

Claude Code no sustituye criterio. Sustituye tecleo, búsqueda, cambios repetidos, revisiones iniciales y parte del trabajo de ejecución.

Si usas bien Claude Code, vas a poder trabajar más rápido, revisar mejor y mantener más control sobre lo que cambia en tu proyecto.

Regla final

Si Claude Code va a tocar varios ficheros, que deje rastro de lo que ha hecho.