

Guía gratuita · IA para construir

Claude Code paso a paso

Claude Code es un entorno de programación asistida por IA que trabaja sobre tus archivos: lee el proyecto, propone cambios, edita, ejecuta comandos y ayuda a revisar resultados. Esta guía explica cómo usarlo con control, incluso si no programas todos los días.

[Descargar guía PDF](#) [Empezar](#) [Qué es](#)

Un asistente técnico dentro de tu proyecto

A diferencia de un chat general, Claude Code puede trabajar en el repositorio: inspecciona archivos, entiende estructura, ejecuta comandos, modifica código y pide aprobación antes de acciones importantes.

[Entender](#)

Lee tu proyecto

Puede explicar carpetas, encontrar archivos importantes, rastrear dónde ocurre una función y resumir cómo está construido algo.

[Construir](#)

Edita con contexto

Puede crear páginas, arreglar errores, ajustar estilos, escribir scripts, modificar componentes y mantener patrones existentes.

[Verificar](#)

Ejecuta pruebas

Puede correr comandos, leer errores, proponer correcciones y repetir hasta que el resultado cargue o pase la validación.

[Uso detallado](#)

Cómo usar Claude Code desde cero

El mejor resultado sale cuando no empiezas pidiendo “hazlo todo”. Primero dale contexto, deja que explore, pide plan, aprueba cambios y revisa con pruebas.

1

Instala y abre Claude Code

Según la documentación oficial, Claude Code puede instalarse con el instalador nativo o con npm. Después entras a la carpeta del proyecto y ejecutas

```
claude
```

.

```
cd ruta/de/tu/proyecto claude
```

- Si es la primera vez, inicia sesión cuando te lo pida.
- En Windows puede usarse con PowerShell, Git Bash o WSL según tu instalación.

2

Empieza pidiendo un recorrido

Antes de editar, pídele que entienda el proyecto. Esto evita cambios a ciegas.

```
Explicame qué hace este proyecto, cuál es la estructura principal y qué archivos debería conocer antes de ed
```

- Útil si descargaste una plantilla.
- Útil si vas a modificar una app que no hiciste tú.

3

Crea memoria del proyecto con CLAUDE.md

La documentación recomienda usar

```
CLAUDE.md
```

para guardar instrucciones del proyecto: comandos, reglas, estilo, rutas importantes y preferencias.

```
/init
```

- Después revisa el archivo generado.
- Agrega cosas como “usar español”, “no tocar estos archivos”, “comando de pruebas” o “estilo visual”.

4

Haz pedidos específicos

Mientras más claro sea el pedido, menos correcciones tendrás. Incluye archivo, objetivo, restricciones y cómo verificar.

```
En la página de guías, actualiza la tarjeta de Claude Code. No borres las demás guías. Luego verifica que el
```

- Evita: “arréglalo”.
- Mejor: describe qué está mal y cómo debe quedar.

5

Pide que explore antes de editar

Para cambios medianos o delicados, pide primero análisis y plan. Después autorizas implementación.

```
Primero revisa los archivos relacionados y dime qué cambiarías. No edites todavía.
```

- Útil para refactors.
- Útil cuando no sabes dónde está el problema.

6

Aprueba cambios con cuidado

Claude Code puede pedir permisos para editar o ejecutar comandos. Revisa qué va a hacer, especialmente si involucra eliminar, mover, instalar o publicar.

- Acepta cambios seguros y concretos.
- Pide explicación si no entiendes el comando.
- No apruebes comandos destructivos sin respaldo.

7

Pide verificación

No basta con que edite. Pide pruebas o comprobaciones: cargar página, correr build, revisar enlaces, ejecutar tests o validar responsive.

Después de editar, abre la página local, verifica que cargue y revisa que no haya enlaces rotos.

- Para código: pide tests.
- Para páginas: pide carga, enlaces y revisión visual.

8

Corrige temprano

Si se va por otro camino, detén la sesión o redirige rápido. La documentación recomienda corregir pronto y limpiar contexto entre tareas diferentes.

- Usa

`Esc`

para detener una acción.

- Usa

`/clear`

entre trabajos no relacionados.

- Usa

`/rewind`

si necesitas volver atrás.

9

Trabaja por entregables pequeños

Divide el trabajo en partes: primero estructura, luego contenido, luego diseño, luego revisión. Esto reduce errores y te permite dirigir mejor.

Primero crea solo la estructura de la página. Cuando la apruebe, desarrollamos el contenido.

10

Cierra con resumen

Al terminar, pide una lista corta de archivos modificados, qué cambió y qué falta revisar.

Resume qué hiciste, qué archivos tocaste y qué debo revisar antes de publicar.

Prompts útiles

Ejemplos listos para copiar a Claude Code

Estos pedidos están escritos con contexto, objetivo y verificación. Puedes adaptarlos a páginas, apps, dashboards o automatizaciones.

Entender proyecto

Recorrido inicial

Explora este proyecto y explícame: 1. Qué hace. 2. Cuáles son los archivos principales. 3. Cómo se ejecuta 1

Página web

Editar una sección

En la página [archivo], ajusta la sección [nombre]. Objetivo: [qué debe comunicar]. Mantén el estilo visual

Error

Depurar

Estoy viendo este error: [pegar error] Primero explícame qué puede estar pasando. Luego localiza el archivo

Refactor

Mejorar sin romper

Revisa este componente y propón una mejora de organización. No cambies comportamiento visible. Mantén nombre

Tests

Agregar pruebas

Agrega pruebas para [archivo o función]. Cubre el caso normal y un caso borde. Sigue el estilo de pruebas ex

Publicar

Antes de lanzar

Haz una revisión antes de publicar: 1. Enlaces rotos. 2. Errores de consola. 3. Texto visible. 4. Responsive

Comandos y hábitos

Comandos que conviene conocer

La disponibilidad exacta puede variar por versión, plataforma y plan, pero estos aparecen en la documentación oficial como parte del flujo de Claude Code.

Inicio

```
/help
```

Muestra comandos disponibles dentro de la sesión.

Proyecto

```
/init
```

Genera un archivo

```
CLAUDE.md
```

inicial para documentar contexto del proyecto.

Contexto

```
/clear
```

Limpia contexto entre tareas distintas para evitar confusión.

Volver

```
/rewind
```

Permite volver a un punto anterior de la conversación o estado.

Permisos

```
/permissions
```

Configura qué acciones puede ejecutar y cuáles requieren aprobación.

Retomar

```
/resume
```

Permite continuar conversaciones anteriores cuando aplica.

Checklist

Antes de dejarlo trabajar solo

Claude Code es potente, pero conviene dirigirlo con límites claros. Este checklist ayuda a mantener control.

Contexto claro

Explicaste objetivo, archivo, resultado esperado y restricciones.

Exploración previa

Le pediste revisar antes de editar si la tarea es delicada.

Permisos revisados

Entiendes qué comandos o ediciones va a ejecutar.

Cambios pequeños

Dividiste el trabajo para revisar por partes.

Verificación

Pediste pruebas, build, carga local o revisión de enlaces.

Resumen final

Sabes qué archivos cambió y qué queda pendiente.

Fuentes consultadas

Documentación oficial

Esta guía está basada en la documentación y ayuda oficial de Claude Code consultada en agosto de 2026.

Quickstart

Instalación, inicio de sesión y primera sesión.

[Ver fuente](#)

Best Practices

Buenas prácticas de contexto, prompts, verificación y sesiones.

[Ver fuente](#)

Commands

Referencia de comandos dentro de Claude Code.

[Ver fuente](#)

Use cases

Casos comunes: entender código, depurar, tests, PRs y skills.

[Ver fuente](#) [Sigue la ruta](#) [Siguiendo guía](#)

Implementar IA en una empresa

Mapear procesos, revisar normativa, priorizar el primer caso de uso, evaluar datos y definir roles.

[Ver guía](#)→

[Ver todas las guías](#)

Siguiente paso

Cuéntame tu caso y lo aterrizamos

Si esta guía te sirvió, el paso siguiente es aplicarla a tu trabajo: una clase personalizada, un tablero, una automatización o un sistema de gestión.

Escríbeme tu caso Agendar clase