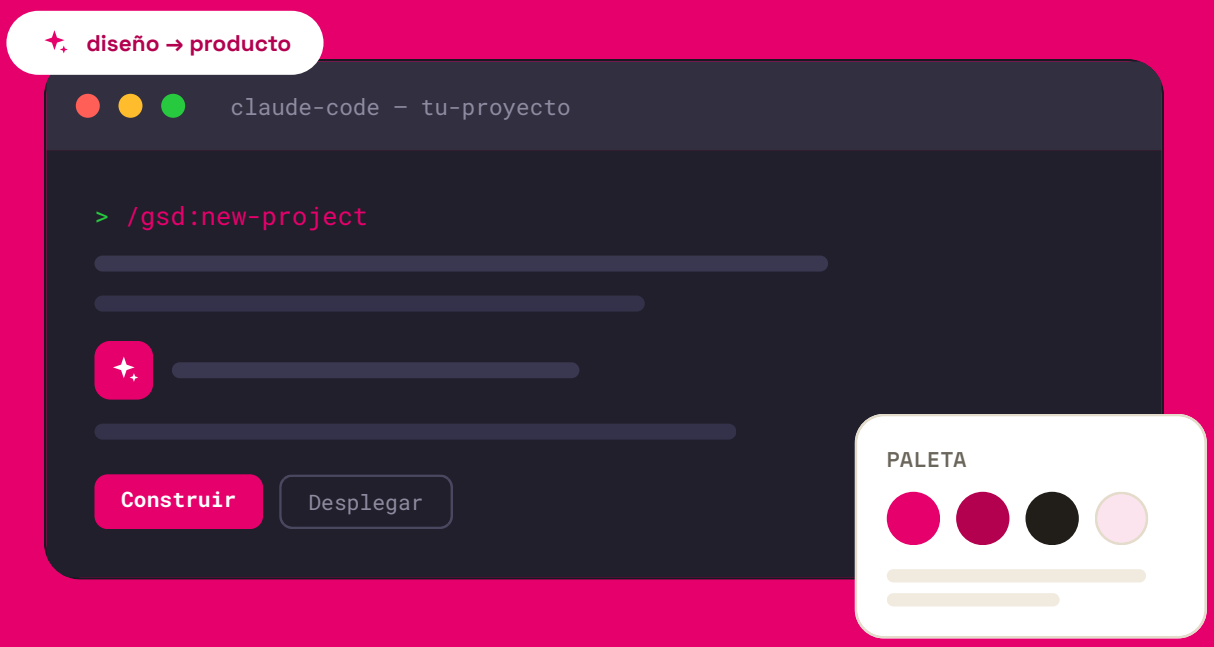
KAOS CONCEPT

MANUAL PRÁCTICO · EDICIÓN EN ESPAÑOL

Claude Code para diseñadores

Diseña, construye y publica productos reales **sin depender de saber programar.** Una guía paso a paso para llevar tus ideas de la pantalla de Figma a una web en producción.



Adaptación libre y reescrita en español, con fines educativos, del ensayo **«Claude Code for Designers: A Practical Guide»** de Tommaso Nervegna.

2026

CONTENIDO

Lo que vas a encontrar

ANTES DE EMPEZAR

Este manual es una adaptación en español pensada para diseñadores. No necesitas experiencia previa programando: si la tienes, mejor todavía. Los nombres de comandos, herramientas y el método GSD son los reales; el resto está reescrito para que cualquiera pueda seguirlo de un tirón.

PUNTO DE PARTIDA

01 El problema de la traducción

Durante años, el consejo para cualquier diseñador que quería ir más allá de la maqueta fue el mismo: **«aprende a programar»**. Suena razonable, pero esquiva el problema de verdad. La fricción no está en escribir líneas de código; está en **traducir** una idea de diseño a algo que funciona de verdad.

Esa traducción es la que duele: montar el entorno, gestionar dependencias, pelearte con frameworks que cambian cada pocos meses y, sobre todo, saltar una y otra vez entre dos formas de pensar muy distintas. Ese salto mental constante —no la sintaxis— es el auténtico cuello de botella.

Aquí es donde las herramientas de IA cambian las reglas. **Claude Code** entiende tu intención de diseño y produce código que funciona, eliminando esa capa de traducción. Tú sigues pensando como diseñador; la máquina se encarga del idioma de las máquinas.

LA IDEA CLAVE

No vas a «convertirte en programador». Vas a ampliar lo que ya sabes hacer como diseñador hasta incluir el último tramo: construir y publicar de verdad.

MOTIVACIÓN

02 Lo que vas a poder construir

Conviene aterrizar esto con ejemplos reales, no con promesas. Con este flujo de trabajo puedes pasar de la idea a algo desplegado y usable en muy poco tiempo:

- Un **agente con IA que rastrea relojes vintage** a la venta, montado en un fin de semana —algo que por la vía tradicional habría supuesto semanas de coordinación.
- Una **canalización de Figma a código** en la que un cambio de diseño regenera los componentes y actualiza la web automáticamente.

La diferencia importante: no son prototipos de mentira ni pantallas que solo simulan funcionar. Son **herramientas reales**, desplegadas en internet, capaces de manejar usuarios y datos de verdad. Pasas de la «maqueta clicable» al **producto vivo**.

Lo que dejas atrás

Maquetas estáticas, prototipos que se rompen, esperar semanas a que alguien «traduzca» tu diseño y descubrir tarde que algo no se podía hacer.

Lo que ganas

Productos funcionales que puedes enseñar, compartir por un enlace y mejorar al instante. Control total sobre cada detalle, de principio a fin.

MANOS A LA OBRA

03 Prepara tu entorno

Necesitas dos piezas: **Claude Code** (el asistente que escribe y ejecuta el código) y **GSD**, un sistema que mantiene a Claude ordenado y enfocado durante proyectos largos. Se instalan en unos minutos.

1

Instala Claude Code

En Mac se instala con un único comando en la terminal. En Windows hay un instalador nativo. Una vez instalado, lo arrancas escribiendo **claude**.

Terminal — macOS

```
$ npm install -g @anthropic-ai/claude-code
$ claude
```

2 Instala GSD (Get Shit Done)

GSD es un sistema de «meta-prompts» que mantiene a Claude centrado, evita que pierda el hilo en proyectos largos, coordina varios agentes a la vez y deja registrada cada decisión.

Terminal

```
$ npx get-shit-done-cc
```

3 Activa el modo autónomo (opcional)

Si quieres que Claude trabaje sin pedirte permiso a cada paso, arráncalo con esta opción. Gana mucha velocidad porque no se detiene a esperar tu visto bueno constantemente.

Terminal

```
$ claude --dangerously-skip-permissions
```

> ⚠ **ÚSALO CON CABEZA**
>
> El modo autónomo es cómodo, pero deja que Claude actúe sin confirmar. Resérvalo para proyectos nuevos donde tú controlas el alcance y no hay nada valioso que se pueda romper sin querer.

FUNDAMENTOS

04 La terminal sin miedo

La terminal asusta porque parece que hay que memorizar cientos de comandos. Buenas noticias: para empezar te bastan **cinco**. El resto del trabajo pesado lo hace Claude por ti.

`claude`	Inicia Claude Code en la carpeta actual
`cd nombre-carpeta`	Entra en una carpeta
`cd ..`	Sube un nivel (vuelve a la carpeta anterior)
`ls` · `dir`	Lista lo que hay en la carpeta (ls en Mac, dir en Windows)
`pwd`	Te dice en qué carpeta estás ahora mismo

TRUCO QUE LO CAMBIA TODO

Usa la **terminal integrada** de un editor como Cursor o VS Code. Así ves tus archivos y la conversación con Claude en la misma pantalla, sin saltar de ventana. Para un diseñador, esto es la diferencia entre sentirse perdido y sentirse en casa.

EL CORAZÓN DEL MÉTODO

05 Tu primer proyecto con GSD

Aquí está el flujo completo. GSD lo divide en pasos claros, cada uno con su propio comando. Vamos a recorrerlos en orden, que es justo como los usarás en la vida real.

5.1 · Arranca el proyecto

El comando **/gsd:new-project** pone en marcha cuatro fases encadenadas que convierten tu idea en un plan accionable:

Claude Code

```
> /gsd:new-project
```

- **Preguntas.** Claude te entrevista: intención de diseño, público, contenido y preferencias técnicas.
- **Investigación** (opcional). Analiza opciones de tecnología, patrones habituales y posibles trampas.
- **Requisitos.** Genera un documento **REQUIREMENTS.md** que separa lo esencial (V1), lo deseable (V2) y lo que queda fuera de alcance.
- **Hoja de ruta.** Trocea el proyecto en fases, cada una con entregables concretos.

5.2 · Discute la fase — el arma secreta

Este es el paso que separa un resultado **con tu voz de diseño** de uno genérico de IA. Antes de construir nada, le cuentas a Claude cómo quieres que se vea y se comporte esa fase.

Claude Code

```
> /gsd:discuss-phase 1
```

Aquí defines espaciados, comportamiento de los **hover**, puntos de quiebre responsive y jerarquía visual. Todo queda guardado en un documento de contexto (por ejemplo, **02-CONTEXT.md**) que evita el típico resultado «de plantilla» y conserva tu criterio en cada decisión.

NO TE SALTES ESTE PASO

Es tentador ir directo a construir, pero invertir cinco minutos describiendo el aspecto y las interacciones es lo que hace que el resultado parezca tuyo y no de cualquiera.

5.3 · Planifica la fase

Con **/gsd:plan-phase**, Claude sigue tres pasos: **investiga** (con varios agentes en paralelo), **planifica** dividiendo el trabajo en tareas pequeñas e independientes, y **verifica**ese plan contra tus requisitos y tu contexto de diseño hasta que pasa todos los chequeos.

Claude Code

```
> /gsd:plan-phase 1
```

5.4 · Ejecuta la fase

/gsd:execute-phase construye de verdad. Las tareas independientes se ejecutan en «olas» paralelas para ahorrar tiempo, y cada una recibe un contexto fresco para que la calidad no se degrade. Tras cada tarea se hace un **commit automático** en Git: un historial limpio y reversible.

Claude Code

```
> /gsd:execute-phase 1
```

¿QUÉ ES UN COMMIT?

Piensa en un commit como un «punto de guardado» en un videojuego. Si algo sale mal, puedes volver atrás a un punto exacto sin perder todo lo demás. Claude los crea por ti, así que no tienes que preocuparte por ello.

5.5 · Verifica el trabajo (la prueba humana)

Por último, **/gsd:verify-work** te pone en el papel de usuario real. Recibes una lista de comprobación para contrastar lo construido con la especificación. Si encuentras algo que falla, Claude lo diagnostica, genera un plan de arreglo y lo ejecuta como tareas nuevas. Es un bucle de validación que se cierra solo.

Claude Code

```
> /gsd:verify-work 1
```

SAL AL MUNDO

06 Publica tu trabajo con Vercel

Tener algo que funciona en tu portátil está muy bien, pero la magia ocurre cuando lo puedes compartir por un enlace. **Vercel** hace ese paso casi trivial y, además, gratis para proyectos personales.

1 **Conecta y despliega**

Crea una cuenta en Vercel enlazada con GitHub, importa tu repositorio y despliega con un clic. A partir de ahí, cada vez que envíes cambios a GitHub, **Vercel vuelve a publicar solo**.

2 **Documenta para tu yo del futuro**

Pídele a Claude dos archivos: un **README.md** que explique el proyecto y cómo arrancarlo, y un **claude.md** que recoja la arquitectura y tus preferencias de diseño. El segundo es oro: le da contexto a Claude en futuras sesiones para que no empiece de cero.

3 **Añade tu propio dominio**

Compra un dominio en Vercel o en cualquier registrador, configura los registros DNS para que apunten a Vercel y espera la propagación: suele tardar entre 5 y 30 minutos. Y ya tienes tu proyecto en una dirección con tu nombre.

EL CICLO QUE ENGANCHA

Editas → Claude construye → haces push → Vercel publica. En cuestión de minutos ves tu cambio en vivo, en una URL real. Ese ciclo tan corto es lo que hace que iterar sea casi adictivo.

SUBIR EL NIVEL

07 De web estática a app real

Una web estática es un gran comienzo, pero quizá quieras ir más allá: usuarios que inician sesión, datos que se guardan, funciones con IA. Para eso necesitas un par de servicios más, y GSD también te acompaña aquí.

Planifica la app

Usa **/gsd:quick** para lanzar preguntas rápidas sobre arquitectura y estrategia de implementación sin montar todo el flujo formal. Es perfecto para tareas sueltas manteniendo la calidad.

Claude Code

```
> /gsd:quick ¿cómo estructuro el login y la base de datos?
```

⚠ **TUS CLAVES, A BUEN RECAUDO**

Las claves de API se guardan en archivos de variables de entorno (**.env**) y se protegen con un archivo **.gitignore** que evita subirlas sin querer a internet. Nunca pegues una clave directamente en el código visible.

Conecta Supabase y la API de Anthropic

- **Supabase** te da base de datos y sistema de inicio de sesión, con un plan gratuito muy generoso.
- El **login con Google** (OAuth) te ahorra todo el lío de gestionar emails y contraseñas a mano.
- La clave de la **API de Anthropic** se guarda en las **Edge Functions** de Supabase, nunca en el código del navegador.
- Pruebas todo en local y, cuando funciona, añades las variables de entorno en Vercel para producción.

Supabase

Base de datos, autenticación y funciones de servidor. El «detrás de la cortina» de tu app.

API de Anthropic

El acceso a Claude para que tu app tenga funciones de IA de verdad, no de adorno.

EL MAPA COMPLETO

08 El ciclo GSD de un vistazo

Si te quedas con una sola imagen de todo este manual, que sea esta. El método GSD es un ciclo: lo recorres una vez por cada fase del proyecto, y cada vuelta deja registro de las decisiones tomadas.

1. **`/gsd:new-project`**
   Preguntas, investigación, requisitos y hoja de ruta. Solo una vez al inicio.
2. **`/gsd:discuss-phase`**
   Defines la estética y las interacciones de la fase. Tu voz de diseño.
3. **`/gsd:plan-phase`**
   Investiga, planifica en tareas pequeñas y verifica el plan contra los requisitos.
4. **`/gsd:execute-phase`**
   Construye en paralelo y guarda commits automáticos tras cada tarea.
5. **`/gsd:verify-work`**
   Pruebas como usuario real y se arregla lo que falle. Cierra el bucle.

Se repite por cada fase: vuelves al paso 2 con la siguiente fase del proyecto. Para tareas sueltas y rápidas, tienes `/gsd:quick`.

Tres principios que lo sostienen

Ingeniería de contexto

Cada tarea recibe contexto fresco, así la calidad no se degrada en proyectos largos.

Commits atómicos

Cambios pequeños y trazables. Volver atrás es fácil y seguro.

Guiado por especificación

Cada decisión se documenta antes de programar. Nada de improvisar a ciegas.

NIVEL AVANZADO

09 Flujo avanzado con Figma

Aquí es donde un diseñador saca toda su ventaja. Gracias al **MCP de Figma**, Claude puede leer tus archivos de Figma directamente. Tu diseño deja de ser una referencia que alguien interpreta: se convierte en la **fuente de verdad** de la que sale el código.

Diseñas en Figma → Claude extrae los cambios → genera componentes → push a GitHub → se despliega solo

La **primera fase** extrae los **design tokens** —colores, tipografía, espaciado— y los lleva al código. Las siguientes construyen componentes y pantallas fieles al píxel, porque los valores salen directos de Figma en lugar de ser aproximados a ojo.

Por qué este flujo funciona tan bien

- Figma sigue siendo tu única fuente de verdad: no tienes que tocar el editor de código para ajustar el diseño.
- Mantienes el control de cada estado, interacción y comportamiento responsive.
- Los valores exactos salen de Figma, así que se acaban los errores de interpretación.
- El resultado es código listo para producción: con TypeScript, accesibilidad y documentación.
- Iteras rápido con interacciones y datos reales, no con maquetas estáticas.

EL SUEÑO DE TODO DISEÑADOR

Cambias un color o un espaciado en Figma, y ese cambio puede llegar al producto en producción sin que nadie lo «traduzca» por el camino. Se acabó el clásico desajuste entre lo diseñado y lo entregado.

LA FOTO GRANDE

10 Por qué esto te importa

Quizá te preguntes: ¿no es esto simplemente «vibe coding», construir a ojo y rezar para que funcione? Justo lo contrario. La estructura de GSD existe para evitar ese caos.

- **Cuestiona suposiciones** y captura los casos límite antes de escribir una línea.
- La **ingeniería de contexto** mantiene la calidad alta incluso en proyectos largos.
- El **desarrollo guiado por especificación** documenta cada decisión antes de programar.
- La **orquestación de varios agentes** mantiene el trabajo ordenado y evita que se degrade.

El resultado para ti: puedes lanzar con calidad de producción de forma **independiente**, probar ideas en entornos reales y controlar la experiencia completa, de la primera pantalla al último detalle.

No se trata de reemplazar a los desarrolladores.

Se trata de ampliar lo que tú, como diseñador, puedes hacer. Las fronteras entre disciplinas eran en parte artificiales; estas herramientas las difuminan. Y si ya traes algo de base técnica, esa intuición se convierte en tu superpoder.

DUDAS RESUELTAS

11 Preguntas frecuentes

¿Necesito saber programar?

Ayuda, pero no es imprescindible. Claude se encarga de la sintaxis. Lo que aportas tú —criterio de diseño, sentido del producto, atención al detalle— es justo lo que la IA no tiene.

Tengo algo de experiencia técnica de hace años. ¿Sirve de algo?

Es la posición ideal. Tu intuición sobre cómo encajan los sistemas sigue siendo válida; Claude se ocupa de la sintaxis moderna que quizá ya no recuerdas. La combinación es muy potente.

Me sale un error y no entiendo nada. ¿Qué hago?

Cópialo y pégalo tal cual en Claude Code. Lo lee, lo diagnostica y normalmente lo arregla. Los mensajes de error, que tanto intimidan, son simplemente más información para Claude.

¿Cuánto cuesta todo esto?

Poco o nada para empezar. Claude Code tiene plan gratuito, Vercel es gratis para proyectos personales y Supabase ofrece un plan gratuito muy generoso. Puedes construir y publicar sin gastar.

¿La calidad del código es buena de verdad?

Suele superar a la de un desarrollador junior, porque aplica buenas prácticas de forma constante y sin despistes. Y como todo queda documentado, es fácil de mantener.

¿Y la seguridad?

Se gestiona con variables de entorno para las claves y con el control de acceso de la base de datos. GSD te guía para hacerlo bien desde el principio, no como un parche al final.

PARA IMPRIMIR

12 Chuleta de comandos

Tu referencia rápida. Recorta esta página, pégala junto al monitor y tenlo todo a mano.

Comandos GSD

`/gsd:help`	Muestra todos los comandos disponibles.
`/gsd:new-project`	Arranca un proyecto: preguntas, investigación, requisitos y hoja de ruta.
`/gsd:discuss-phase N`	Define estética e interacciones de la fase N (el arma secreta).
`/gsd:plan-phase N`	Investiga, planifica en tareas atómicas y verifica el plan.
`/gsd:execute-phase N`	Construye la fase N en paralelo, con commits automáticos.
`/gsd:verify-work N`	Prueba humana (UAT): contrastas el resultado con la especificación.
`/gsd:quick`	Tareas sueltas y rápidas manteniendo la calidad.
`/gsd:complete-milestone`	Archiva un hito completado.

Terminal básica

`claude`	Inicia Claude Code.
`cd carpeta`	Entra en una carpeta.
`cd ..`	Sube un nivel.
`ls · dir`	Lista archivos (ls en Mac, dir en Windows).
`pwd`	Muestra en qué carpeta estás.
`npx serve`	Previsualiza el sitio en local (http://localhost:3000).

Instalación y despliegue

`npm install -g @anthropic-ai/claude-code`	Instala Claude Code (Mac).
`npx get-shit-done-cc`	Instala el sistema GSD.
`claude --dangerously-skip-permissions`	Arranca en modo autónomo.

PARA TERMINAR

El futuro del diseño es construir

Las herramientas de IA no vienen a reemplazar a diseñadores ni a desarrolladores. Vienen a derribar las fronteras artificiales entre disciplinas. Claude por sí solo no basta: necesita la estructura de GSD para ser fiable y consistente. Juntos, te dan algo que hasta hace poco era impensable para alguien que no programa a diario.

El futuro del diseño no es solo entregar maquetas estáticas y esperar a que alguien las haga realidad. Es construir sistemas que funcionan, ponerlos en manos de usuarios reales y aprender de esa experiencia para mejorar. Si ya traes algo de base técnica, apóyate en ella. Y si no, no pasa nada: este es el mejor momento de la historia para empezar.

Ya no diseñas lo que otros construirán.
Diseñas y construyes tú.