GUÍA NODESTUDIO

Claude diseña mejor

42 skills para que Claude deje de diseñar genérico, ordenadas en capas y con un instalador.

⏱ Instalación del kit: 49 segundos en nuestra prueba

1 Instalá el kit con Claude Code

No instales las 42: Claude carga las skills que tiene a mano y demasiadas se pisan. Empezá con el kit de 3 y sumá las que necesites. El instalador lee un catálogo con las 42, dónde está cada una y qué necesita, y hace todo por vos.

CATÁLOGO recursos.valentinpanizza21.workers.dev/diseno/catalogo.md

Qué necesitás

- **Claude Code instalado.** No viene en el plan gratis de Claude: necesitás Pro o uno superior.
- **Node.js**, para Playwright (Claude Code te avisa si no lo tenés).
- **Internet**, para bajar los repositorios de GitHub.

Todas son gratis. Algunas necesitan programas aparte (Figma, Blender, After Effects): está en las páginas 4 y 5.

PROMPT PARA PEGAR EN CLAUDE CODE

```
Quiero instalar skills de diseño para Claude Code. El catálogo está acá:
https://recursos.valentinpanizza21.workers.dev/diseno/catalogo.md

Hacé todo vos, sin pedirme que corra comandos, y al final contame en palabras simples qué hiciste.

1. Leé el catálogo entero.
2. Instalá las 3 marcadas KIT y además estas: [números del catálogo, o "ninguna"].
3. Para cada skill: bajá su repositorio en una carpeta temporal (con git, o el ZIP de GitHub si no hay git) y copiá la carpeta de la ruta indicada, con el nombre del catálogo, a mi carpeta de skills personales de Claude Code (~/.claude/skills en Mac y Linux, %USERPROFILE%\.claude\skills en Windows). Si ya existe, no la pises: avisame.
4. Para los MCP, corré el comando del catálogo. Para los plugins, decime los comandos /plugin que tengo que escribir yo.
5. Comprobá que cada skill quedó con su SKILL.md y mostrame una lista: qué se instaló, dónde y qué necesita aparte según el catálogo.
6. Borrá las carpetas temporales. No toques otras skills ni otros MCP.

Al terminar, recordame cerrar y volver a abrir Claude Code.
```

2 Usá el kit de 3

Las tres juntas cubren lo que más falla cuando Claude diseña: el gusto, la terminación y que no mira lo que hizo.

Una base de gusto `frontend-design`

PEDÍSELO ASÍ

```
Usá la skill frontend-design para armar en index.html la landing de [tu negocio]: [secciones que querés].
```

En la prueba (una panadería inventada): eligió una paleta y una tipografía propias en vez del look genérico, puso un detalle animado en el hero y el horario con "Abierto ahora" calculado con la hora real.

Auditar y pulir `impeccable`

PEDÍSELO ASÍ

```
Usá impeccable con el comando audit sobre index.html y aplicá las 3 correcciones más importantes.
```

En la prueba encontró un título de 140 px que empujaba el botón fuera de la pantalla, links de 43 px (el mínimo táctil es 44) y una tipografía sobreusada, y los corrigió. Otros comandos: `/impeccable polish`, `/impeccable typeset`, `/impeccable animate` (la "animate" del carrusel).

Que mire lo que hizo `playwright-mcp`

PEDÍSELO ASÍ

```
Abrí index.html en el navegador con Playwright, sacá una captura en celular y en compu, y corregí lo que se vea mal.
```

Sin esto, Claude diseña a ciegas: escribe el código y no ve el resultado.

2 Las 42, por capa (el número es el que va en el prompt)

Capa 1 · Frontend y UI

#	Skill	Qué hace	Necesita aparte
1	`frontend-design` · kit	Layouts con dirección estética fuerte, no plantillas	Nada
2	`impeccable` · kit	24 comandos para auditar, pulir, tipografía, color y layout	Nada
3	`taste-skill`	Dirección de arte con referencias y chequeo en el navegador	Nada
4	`animate`	Microtransiciones con criterio (comando de impeccable)	Impeccable instalada
5	`motion-art-direction`	Define el lenguaje de movimiento antes de animar	Nada
6	`theme-factory`	Temas y paletas listos para aplicar a lo que armes	Nada
7	`figma-design-to-code`	De un diseño de Figma a código fiel	MCP de Figma conectado
8	`playwright-mcp` · kit	Claude abre el navegador, mira lo que hizo y lo corrige	Node.js
9	`ui-ux-pro-max`	Estilos, paletas, tipografías y reglas de UX para buscar	Python 3 para sus búsquedas
10	`designer-skills`	63 skills de diseño en 8 plugins, de investigación a handoff	Nada (colección de 8 plugins)

Capa 2 · Imagen, gráficos y video

#	Skill	Qué hace	Necesita aparte
11	`nano-banana`	Imágenes desde la terminal con Gemini	Gemini CLI con la extensión nanobanana y una clave de Gemini con facturación activa
12	`canvas-design`	Pósters, PNG y PDF con diseño de verdad	Nada
13	`algorithmic-art`	Arte generativo con código (p5.js)	Nada
14	`remotion`	Buenas prácticas para hacer video con React	Node.js y un proyecto de Remotion
15	`blender-mcp`	Claude arma escenas 3D en Blender	Blender con el addon del repo, y uv
16	`after-effects`	Automatiza After Effects con scripts	Mac con After Effects (pago)

2 Las 42, por capa (continuación)

Capas 3 y 4 · Producto e interacción

#	Skill	Qué hace	Necesita aparte
17	`figma-generate-design`	Pantallas de app directo en Figma	MCP de Figma conectado
18	`presentation-deck`	Presentaciones de diseño para mostrar tu trabajo	Nada
19	`micro-interaction-spec`	Microinteracciones bien definidas	Nada
20	`conversational-ux`	Conversaciones (chats, bots) que no se rompen	Nada
21	`error-handling-ux`	Errores que no frustran al usuario	Nada
22	`feedback-patterns`	Respuesta clara a cada acción	Nada

Las 20 de reserva

#	Skill	Qué hace	Necesita aparte
23	`figma-implement-motion`	Animaciones de Figma pasadas a código	MCP de Figma conectado
24	`figma-generate-library`	Design system en Figma desde tu código	MCP de Figma conectado
25	`figma-code-connect`	Conecta componentes de Figma con tu código	MCP de Figma conectado
26	`figma-use-slides`	Presentaciones en Figma Slides	MCP de Figma conectado
27	`figma-generate-diagram`	Diagramas en FigJam	MCP de Figma conectado
28	`figma-shaders`	Fondos y efectos con shaders en Figma	MCP de Figma conectado
29	`web-artifacts-builder`	Apps web completas con React y shadcn	Node.js
30	`webapp-testing`	Prueba tu interfaz en un navegador real	Python 3 con Playwright
31	`slack-gif-creator`	GIFs animados que pesan poco	Python 3
32	`brand-guidelines`	Aplica una guía de marca a todo lo que hace	Nada (trae la marca de Anthropic como ejemplo: cambiala por la tuya)
33	`logo-animation`	Animación de logos	Nada
34	`color-motion`	Color que se mueve con sentido	Nada
35	`shot-composition`	Composición de cada plano de un video	Nada
36	`remotion-captions`	Subtítulos animados en Remotion	Un proyecto de Remotion
37	`design-token`	Tokens de diseño ordenados	Nada
38	`accessibility-audit`	Auditoría de accesibilidad	Nada
39	`typography-scale`	Escala tipográfica coherente	Nada
40	`spacing-system`	Sistema de espaciado	Nada
41	`dark-mode-design`	Modo oscuro bien hecho	Nada
42	`ux-writing`	Textos de interfaz claros	Nada

3 Lo que nos pasó al probarlas

EL MCP QUEDÓ "ESPERANDO APROBACIÓN"

Si Playwright se agrega solo para un proyecto, Claude Code pide aprobarlo la primera vez que lo abrís en esa carpeta. El instalador lo agrega para tu usuario (`-s user`) y así no lo pide. Si igual te aparece, abrí Claude Code y aceptalo.

PONE DATOS DE EJEMPLO

En la landing de prueba dejó un número de WhatsApp de mentira y horarios inventados, y lo avisó al final. Revisá siempre los datos antes de publicar.

EL DETECTOR DE IMPECCABLE A VECES EXAGERA

Marcó "padding apretado" en una sección que tenía espacio de sobra. Claude lo revisó en el navegador y lo dejó como falso positivo. Si una corrección no te convence, pedile que te muestre la captura.

ALGUNAS NECESITAN PROGRAMAS APARTE

Las de Figma necesitan el MCP de Figma conectado. after-effects funciona solo en Mac con After Effects. nano-banana necesita una clave de Gemini con facturación activa. blender-mcp necesita Blender con su addon. Están en la columna "Necesita aparte".

CÓMO ARMAMOS LA LISTA

Verificamos las 42 contra GitHub el 20/10/2026: cada una existe en la ruta del catálogo. Si un repo cambia, el instalador te avisa que no encontró la carpeta.



NodeStudio

Automatizamos procesos de negocio con IA y sistemas a medida.

nodestudioai.com