

Matt Pocock Skills — los 16 skills del ingeniero que enseña TypeScript a medio internet

CATEGORÍA · SKILLS Y COMANDOS

Matt Pocock (creador de Total TypeScript, el que enseña TypeScript a medio internet) subió un con 16 skills para Claude Code que te ayudan a construir proyectos en serio en vez de vibe codear. El ya tiene 50.2k estrellas en GitHub, licencia MIT, y se instala con un solo comando: ``npx skills@latest a`

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

Matt Pocock (creador de Total TypeScript, el que enseña TypeScript a medio internet) subió un con 16 skills para Claude Code que te ayudan a construir proyectos en serio en vez de vibe codear. El ya tiene 50.2k estrellas en GitHub, licencia MIT, y se instala con un solo comando: ``npx skills@latest a`

PARA QUÉ SIRVE

Matt Pocock (creador de Total TypeScript, el que enseña TypeScript a medio internet) subió un con 16 skills para Claude Code que te ayudan a construir proyectos en serio en vez de vibe codear. El ya tiene 50.2k estrellas en GitHub, licencia MIT, y se instala con un solo comando: ``npx skills@latest add mattpocock/skills``. Acá traduzco al español como si tuvieras 5 años qué hace cada uno y profundizo en los 3 que más se usan: Grill Me te interroga antes de codear hasta que sabes exactamente qué construir, Caveman hace que Claude responda corto y te ahorra hasta 75% de tokens, y Diagnose va paso a paso buscando dónde se rompió un bug (reproduce, aísla, hipotetiza, instrumenta, arregla y deja test de regresión).

DÓNDE APLICARLO

Nivel Intermedio. Temas: matt-pocock-skills, claude-code. Incluye el código de mattpocock/skills.

CÓMO IMPLEMENTARLO

- 1 Ruta rápida: pégale 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

Matt Pocock (creador de Total TypeScript, el que enseña TypeScript a medio internet) subió un con 16 skills para Claude Code que te ayudan a construir proyectos en serio en vez de vibe codear. El ya tiene 50.2k estrellas en GitHub, licencia MIT, y se instala con un solo comando: ``npx skills@latest add mattpocock/skills``.

Acá traduzco al español como si tuvieras 5 años qué hace cada uno y profundizo en los 3 que más se usan: Grill Me te interroga antes de codear hasta que sabes exactamente qué construir, Caveman hace que Claude responda corto y te ahorra hasta 75% de tokens, y Diagnose va paso a paso buscando dónde se rompió un bug (reproduce, aísla, hipotetiza, instrumenta, arregla y deja test de regresión).

Los otros 13 cubren TDD vertical-slice, triage de issues con máquina de estados, conversión de conversaciones a PRDs y a issues de GitHub, mejora de arquitectura, hooks de Claude Code que evitan `git push --force`, migración a `@total-typescript/shoehorn`, scaffolds de cursos y pre-commit con Husky. Te dejo el comando exacto para instalarlos, los 3 pasos del setup y 4 prompts copy-paste para arrancar.

INSTALACIÓN

skills

Colección de skills de Matt Pocock; incluye /grill-me, que te interroga antes de construir.

EL CÓDIGO ESTÁ AQUÍ

<https://github.com/mattpocock/skills>
Licencia MIT · Shell

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/mattpocock/skills>

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/mattpocock/skills.git
cd skills

# 2. Da permisos al instalador si lo trae
chmod +x install.sh && ./install.sh

# 3. Deja la skill donde Claude la lee
mkdir -p ~/.claude/skills
cp -r . ~/.claude/skills/skills

# 4. Comprueba que quedo cargada (dentro de Claude Code)
/context
```

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

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: Skills y comandos

Nivel: Intermedio

Licencia: MIT

Lenguaje: TypeScript

Temas: matt-pocock-skills, claude-code

Código: <https://github.com/mattpocock/skills>

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)