

Anti Hackers. Audita tu app antes de publicar.

CATEGORÍA · SEGURIDAD Y BUENAS PRÁCTICAS

Claude Security es el plugin de Anthropic que pone un equipo de agentes a atacar tu propia app, comprueba cada falla antes de decírtela, y te escribe los arreglos. Esta guía te lleva de la mano aunque nunca hayas abierto una terminal.

QUÉ VAS A ENCONTRAR

- 01 El modo fácil: que Claude te acompañe
- 02 Qué es una terminal y como abrirla
- 03 La instalación explicada línea por línea
- 04 Como leer el reporte sin ser programador
- 05 Qué hacer cuando algo te falla
- 06 Las siete fallas más comunes en apps hechas con IA

FICHA DEL RECURSO

Qué es, para qué sirve y cómo se implementa

FUNCIÓN

Claude Security es el plugin de Anthropic que pone un equipo de agentes a atacar tu propia app, comprueba cada falla antes de decírtela, y te escribe los arreglos. Esta guía te lleva de la mano aunque nunca hayas abierto una terminal.

PARA QUÉ SIRVE

Esto revisa las puertas. Imagina que tu app es un edificio de oficinas. Tiene puertas por donde entra gente de la calle: el formulario de registro, el botón de pagar, la búsqueda, el login. Cada una de esas puertas debería tener su llave y su guardia.

DÓNDE APLICARLO

Cubre: El modo fácil: que Claude te acompañe; Qué es una terminal y como abrirla; La instalación explicada línea por línea; Como leer el reporte sin ser programador; Qué hacer cuando algo te falla.

CÓMO IMPLEMENTARLO

- 1 Empecemos Por Aquí: Tu app es un edificio.
- 2 Antes De Seguir: Esta guía tiene un modo fácil.
- 3 Como Trabaja: Seis pasos. Cien agentes.
- 4 Antes De Empezar: Cinco cosas y nada más.
- 5 Paso 1: Abre la terminal.
- 6 Así Se Ve: Esto es todo lo que vas a hacer.
- 7 Paso 2: Llévala hasta tu proyecto.

EMPECEMOS POR AQUÍ

Tu app es un edificio.

Esto revisa las puertas. Imagina que tu app es un edificio de oficinas. Tiene puertas por donde entra gente de la calle: el formulario de registro, el botón de pagar, la búsqueda, el login. Cada una de esas puertas debería tener su llave y su guardia.

- Cuando armas una app rápido, con Lovable o con IA, esas puertas quedan puestas pero casi nunca revisadas. Funciona, se ve bien, y la publicas. Nadie comprobó si alguna quedó abierta.
- Un auditor de seguridad es la persona que va puerta por puerta probando si abren sin llave. Cobra entre tres y diez mil dólares por hacerlo.
- Claude Security hace ese recorrido por ti. Pone un equipo de agentes de IA a probar cada puerta de tu app, y te entrega una lista de las que abrieron, con la dirección exacta de cada una.

- No necesitas saber programar para usarlo. Necesitas saber abrir una ventana y copiar siete líneas. Eso es lo que vas a aprender aquí.

ANTES DE SEGUIR

Esta guía tiene un modo fácil.

Si en algún punto te pierdes, no tienes que resolverlo solo. Esta guía la puede leer Claude por ti y acompañarte paso a paso, en tu propio ritmo.

- Abre Claude, en la app o en la web. Arrastra este PDF a la conversación, o tócalo con el botón de adjuntar.
- Pégale el texto de abajo. Con eso Claude lee la guía completa, entiende lo que quieres hacer, y te va llevando de un paso al siguiente sin adelantarse.
- Si algo te falla, pégale el error tal cual, con captura o copiado. Claude lo lee y te dice qué pasó y como resolverlo.
- Esto no reemplaza leer la guía. Pero si nunca abriste una terminal, es la diferencia entre trabarte en el paso dos y llegar al final.

PÉGALE ESTO A CLAUDE

Te comparto una guia en PDF. Quiero instalar y correr lo que explica, paso a paso. Leela completa y guiame de a un paso a la vez, en lenguaje simple, sin asumir que se de programacion. Espera mi confirmacion antes de pasar al siguiente paso. Si algo me falla, ayudame a resolverlo antes de seguir.

COMO TRABAJA

Seis pasos. Cien agentes.

No es un programa que busca palabras sospechosas en tu código. Es un equipo que rehace el camino que haría un ladrón de verdad, paso por paso, y solo te dice lo que puede probar. Cada paso alimenta al siguiente. El panel decide qué llega al reporte.

ANTES DE EMPEZAR

Cinco cosas y nada más.

Revisa esta lista antes de tocar nada. Si te falta algo, resuélvelo primero, porque después no vas a saber por qué no funciona.

- Claude Code instalado. Es la versión de Claude que trabaja con los archivos de tu computadora. Si ya lo usas para programar, ya lo tienes. Si no, se descarga desde claude.com/code.
- Un plan de pago de Claude, Pro o Max. El escaneo consume del mismo plan que ya usas.
- Python 3.9 o más nuevo. El plugin lo usa por dentro para contar los votos de los jueces. En Mac ya viene instalado. Si te da error más adelante, se descarga gratis de python.org.
- La carpeta de tu proyecto ubicada. Necesitas saber dónde están los archivos de tu app en tu computadora. Si tu app está en Lovable, en el menú de Lovable puedes exportar el proyecto a tu máquina.
- Y un consejo antes de arrancar. Si es tu primera vez, no lo pruebes en tu app de producción. Prueba en una copia. Así si algo sale raro, no pasa nada.

PASO 1

Abre la terminal.

No muerde. La terminal es una ventana donde le escribes órdenes a tu computadora con palabras, en vez de hacer clic en botones. Es como pedir en un restaurante: en vez de señalar el menú con el dedo, dices el plato en voz alta. Se ve intimidante porque es texto sobre fondo negro, pero solo vas a escribir siete líneas en toda la guía. 1 Si tienes Mac. Presiona la tecla Command y la barra espaciadora al mismo tiempo. Se abre un buscador en el centro de la pantalla. Escribe Terminal y dale Enter. Se abre una ventana con texto. 2 Si tienes Windows. Presiona la tecla de Windows, escribe PowerShell y dale Enter. Es la misma idea con otro nombre. 3 Vas a ver una línea con el nombre de tu computadora y un símbolo al final, que puede ser un signo de porcentaje o de mayor que. Ahí es donde escribes. Ese símbolo ya está puesto, tú no lo escribes. 4 Escribes tu orden y das Enter. Siempre Enter al final, si no, no pasa nada. Y si te equivocas al escribir, borras con la tecla de borrar como en cualquier lado.

ASÍ SE VE

Esto es todo lo que vas a hacer.

Todo lo que vas a escribir se parece a esto. Lo demás lo escribe la computadora. En esta imagen, lo verde es lo que tú escribes, y lo gris es lo que ya está ahí o lo que responde la máquina. Escribes, das Enter, esperas. Ese es el ciclo completo.

PASO 2

Llévala hasta tu proyecto.

La terminal siempre está parada en algún lugar de tu computadora, como si fuera una persona de pie dentro de una carpeta. Ahora mismo está en tu carpeta de usuario. Tienes que caminarla hasta donde viven los archivos de tu app, porque solo va a revisar lo que tenga enfrente. 1 Escribe las letras `cd`, luego un espacio. Nada más todavía. Esas dos letras significan cambiar de carpeta. 2 Ahora abre el Finder en Mac, o el Explorador en Windows, y busca la carpeta de tu proyecto. 3 Arrastra esa carpeta con el mouse y suéltala encima de la ventana de la terminal. La dirección completa se escribe sola. Este es el truco que casi nadie enseña y te ahorra escribir la ruta a mano. 4 Da Enter. Vas a ver que el nombre de tu carpeta ahora aparece en la línea. Eso significa que la terminal ya está adentro. 5 Ahora escribe la palabra `claude` y da Enter. Claude Code arranca dentro de esa carpeta. A partir de aquí ya no estás hablándole a la computadora, le estás hablando a Claude.

ESCRIBE ESTO

```
cd (espacio, y arrastra tu carpeta aquí)
claude
```

PASO 3

Instala el plugin.

Cuatro líneas. Un plugin es una extensión, como cuando le instalas una app a otra app. Estas cuatro líneas se escriben adentro de Claude, no en la terminal pelada. Escribe una, espera que termine, y sigue con la siguiente. 1 La primera línea le dice a Claude de dónde bajar los plugins oficiales de Anthropic. Es como agregar una tienda a tu teléfono. Si ya la tenías, te va a decir que la actualizó. 2 La segunda instala Claude Security, el que hace el escaneo profundo. Te va a mostrar una pantalla pidiendo confirmación, con varias opciones. Elige la primera, que dice instalar para ti, y da Enter. 3 La tercera instala un segundo plugin gratis que se llama security guidance. Ese trabaja solo, en silencio, avisándote cuando escribes algo riesgoso. Te lo recomendando. Misma pantalla, misma opción. 4 La cuarta activa los dos en la sesión. Si te dice que recargó dos plugins, ya está. 5 Si algo falla aquí, lo más probable es que tu Claude Code esté viejo. Necesitas la versión 2.1.154 o más nueva.

ESCRIBE ESTO

```
/plugin marketplace add anthropics/claude-plugins-official  
/plugin install claude-security@claude-plugins-official  
/plugin install security-guidance@claude-plugins-official  
/reload-plugins
```

LA PANTALLA QUE VERÁS

Al instalar te pregunta esto.

Después de la segunda y de la tercera línea aparece esta pantalla. Te está preguntando para quién instalas el plugin. Muévete con las flechas y confirma con Enter. Siempre la primera opción. Las otras dos lo dejan encerrado en una sola carpeta.

PASO 4

Dos ajustes y quedas listo.

Estos dos no son opcionales. El primero evita que el escaneo se detenga cada treinta segundos a pedirte permiso. El segundo hace que el reporte quede atado al código exacto que revisó. 1 El modo automático. Presiona las teclas Shift y Tab juntas, varias veces, hasta que abajo de la ventana diga auto mode on. Sin esto, el escaneo se va a parar en cada paso preguntándote si puede seguir, y son cientos de pasos. 2 El historial de cambios, que se llama git. Es un sistema que guarda fotos de tu proyecto cada vez que avanzas, como el control de cambios de Word pero para toda la carpeta. El escaneo completo funciona sin esto, pero para revisar solo lo que cambiaste, o para que te escriba los arreglos, sí hace falta. 3 Si tu proyecto no lo tiene, dile esto a Claude tal cual, en español: prepara esta carpeta con git y guarda una primera versión. Él lo hace solo y te explica qué hizo. 4 Cuando termines, revisa que tu proyecto no tenga cambios sin guardar. Si los tiene, el reporte te lo va a marcar, y es mejor guardar primero para que el reporte describa el código real.

PASO 5

Córrelo. Y déjalo trabajar.

Ahora sí, el momento. Escribe el comando y sigue las preguntas. Son tres, y todas tienen una opción de no sé que el plugin resuelve por ti. 1 Escribe `/claude-security` y da Enter. Se abre un menú con tres opciones. 2 Elige la primera, escanear el código. Es la que revisa tu app completa, y la que quieres la primera vez. 3 Te va a preguntar qué parte revisar. Si tu proyecto es chico, dile todo. Si es grande, señálale la carpeta donde está la parte que recibe información de afuera, normalmente se llama api, server, backend o src. 4 Te va a preguntar el nivel de esfuerzo. La primera vez elige bajo, para ver como funciona sin gastar mucho. Cuando vayas a publicar de verdad, medio. En la página siguiente te explico los cuatro. 5 Y ahora espera. Puede tomar de dos a treinta minutos según el tamaño. Si quieres ver a los agentes trabajando en vivo, escribe `/workflows`. Puedes cerrar esa vista con Esc y el escaneo sigue corriendo.

ESCRIBE ESTO

```
/claude-security
```

LO QUE VAS A VER

Eliges una vez.

Después trabaja solo. Arriba, el menú que aparece al escribir el comando. Abajo, lo que ves si escribes `/workflows` mientras corre. Puedes cerrar esa vista con Esc y el escaneo sigue. Puedes irte a hacer otra cosa. Te avisa cuando termina.

LOS CUATRO NIVELES

Aquí se decide cuánto gastas.

El nivel no cambia qué tan inteligente es el análisis. Cambia cuánta gente pone a trabajar. Y ojo con esto, los jueces que comprueban cada falla son siempre tres, en todos los niveles. O sea que un escaneo barato encuentra menos cosas, pero las que encuentra están igual de comprobadas.

- Bajo. Un investigador revisando todo, más los tres jueces. Termina rápido y cuesta poco. Úsalo para probar que quedó bien instalado, y para revisar cambios del día a día.
- Medio. El equipo completo, con inventario, mapa de amenazas, un investigador por cada área y una pasada final. Es el que viene por defecto y el que debes usar antes de publicar.
- Alto. Lo mismo pero con el doble de investigadores. En un proyecto grande esto son cientos de dólares de tu plan. No lo corras por curiosidad.
- Máximo. Agrega un equipo que intenta desmentir cada falla que ya pasó los jueces. Resérvalo para código que mueve dinero o guarda datos médicos o legales.

- Mi recomendación honesta: bajo para el día a día, medio antes de publicar, y los otros dos solo cuando hay algo grande en juego.

LOS TRES JUECES

Por qué el reporte es tan corto.

Esta es la parte que hace que valga la pena. Cuando un investigador cree que encontró una falla, no te la dicen. Primero pasa por tres jueces distintos, y cada uno la ataca por un lado diferente. Lo que llega al reporte no es una sospecha. Es un camino comprobado.

EL REPORTE

Tres archivos.

Solo lees uno. Cuando termina, aparece una carpeta nueva dentro de tu proyecto con la fecha y la hora en el nombre. Adentro hay tres archivos y nada más.

- El que termina en RESULTS.md es el que tú abres. Se lee como un documento normal, en cualquier editor de texto. Cada falla viene con qué tan grave es, como la explotaría alguien, y qué hacer para arreglarla.
- El que termina en .jsonl es el mismo contenido pero para máquinas. Ignóralo, salvo que tengas un programador conectándolo a otra herramienta.
- El que dice REVISION es el sello de la corrida: qué versión de tu código se revisó, con qué nivel, y cuántas fallas de cada gravedad. Si tenías cambios sin guardar, el nombre del archivo lo dice.
- Busca la sección que dice cobertura. Ahí te dice qué carpetas quedaron fuera y por qué. Eso importa, porque un reporte limpio no significa que todo esté bien si la mitad no se revisó.
- Esa carpeta se protege sola para que no se te suba por accidente a internet junto con tu código. Si quieres compartirla con alguien, mándala aparte.

ASÍ SE VE UNA FALLA

Un pedazo de un reporte real.

Esta es una falla del escaneo que corrí en la app de prueba. Te dice el daño, el archivo y la línea exacta, como se explotaría, y que los tres jueces la confirmaron. Cada falla del reporte trae estas cuatro cosas. Ni una más.

LOS ARREGLOS

Te los escribe.

No los aplica. Cuando corres /claude-security otra vez y eliges la tercera opción, toma las fallas del reporte y te escribe el arreglo de cada una. Un arreglo así se llama parche, y funciona como una corrección marcada en un documento: está propuesta, y tú decides si la aceptas.

- Nada de eso se hace sobre tu proyecto. Hace una copia aparte, arregla ahí, y tu carpeta original no se toca en ningún momento.
- Después otro agente distinto revisa ese arreglo, corre las pruebas de tu proyecto, y lo mira por su cuenta buscando problemas nuevos que el arreglo pudo crear.
- Solo te entrega el parche si puede responder por tres cosas: que arregla esa falla, que no crea una nueva, y que tu app sigue funcionando igual que antes. Si no puede responder por las tres, te entrega una nota explicando por qué en vez del arreglo.
- Los archivos quedan en una subcarpeta que dice patches, uno por falla. Para aplicar uno, lo más simple es decirle a Claude en español: aplica el parche F1 y explícame qué cambió. Si prefieres el comando, es el de abajo.
- El plugin nunca sube nada a internet, ni guarda cambios, ni toca tu app publicada. Todo se queda en tu computadora esperando tu decisión.

EL COMANDO, SI LO PREFIERES

```
git apply CLAUDE-SECURITY-<fecha>/patches/F1.patch
```

SI ALGO SALE MAL

Los cinco errores que vas a tener.

Estos son los que aparecen siempre la primera vez. Ninguno es grave y todos se resuelven en un minuto.

- Dice `command not found: claude`. Significa que Claude Code no está instalado en tu computadora, o que la terminal no lo encuentra. Instálalo desde claude.com/code y cierra y vuelve a abrir la terminal.
- Dice `not a git repository`. Le falta el historial de cambios a tu carpeta. Escríbele a Claude: prepara esta carpeta con git y guarda una primera versión.
- Escribes `/claude-security` y dice comando desconocido. Los plugins no se cargaron. Cierra Claude por completo, ábrelo de nuevo, y vuelve a intentar. Si sigue igual, corre `/reload-plugins`.
- El escaneo se detiene todo el tiempo pidiendo permiso. No activaste el modo automático. Presiona Shift y Tab hasta que abajo diga `auto mode on`.
- Te avisa que va a consumir muchos tokens. Es real. Cancela, y vuelve a correrlo eligiendo nivel bajo, o señalándole una sola carpeta en vez del proyecto entero.
- Y si te aparece algo que no está en esta lista, pégaselo a Claude tal cual, con el prompt de la página tres. Para eso está.

LAS SIETE FALLAS

Lo que aparece en apps hechas con IA.

Estas siete son las que más salen. Todas aparecieron en la app de prueba que escaneé. Te las explico sin palabras técnicas para que reconozcas la tuya.

- La puerta sin guardia. La dirección de tu app termina en un número. Cambias el uno por el dos y te sale la ficha de otro cliente. Como un hotel que te da la llave de cualquier cuarto sin preguntarte en cuál te quedas.
- El formulario obediente. Tu app le pregunta a la base de datos usando el texto que escribió el usuario, tal cual. Entonces el usuario puede escribir una orden en vez de un nombre, y la base le obedece.
- La llave bajo el tapete. La clave de tu pasarela de pagos o de tu base de datos, escrita dentro de un archivo del proyecto. Cualquiera que vea el código, la tiene.
- Las contraseñas a la vista. Si alguien entra a tu base, se lleva las contraseñas de todos tus usuarios y las puede leer. Deberían estar cifradas y no lo están.
- El aviso de pago falso. Tu app le cree a cualquiera que le diga que un cliente pagó, sin verificar que el aviso venga de verdad de la pasarela de pagos.
- La puerta de servicio abierta. Una dirección que devuelve la lista completa de tus clientes de una sola vez, que tú hiciste para exportar datos y quedó sin llave.
- El error que habla demasiado. Cuando algo falla, tu app le muestra al usuario el mensaje técnico

completo, y ese mensaje le explica al atacante como está construida tu base de datos.

ANTES DE PUBLICAR

El checklist.

Ocho puntos. Repasa esto antes de que tu app salga al mundo, y después cada vez que le hagas un cambio importante.

- Instalaste los plugins y corriste `/reload-plugins`.
- Activaste el modo automático con Shift y Tab, y abajo dice auto mode on.
- Tu proyecto tiene el historial de git y no tiene cambios sin guardar.
- Corriste el escaneo en nivel medio, no en bajo, antes del primer lanzamiento.
- Leíste la sección de cobertura y sabes qué quedó fuera.
- Revisaste cada arreglo antes de aplicarlo. La etiqueta que trae es verificado, nunca probado.
- Cambiaste toda llave que apareció en el código. Moverla de lugar no sirve, si estuvo expuesta hay que reemplazarla.
- Te anotaste volver a correrlo cada tanto. Dos escaneos del mismo código pueden encontrar cosas distintas, y encuentran más con el tiempo. Correrlo seguido es lo que de verdad te cubre.

SIGUIENTE PASO

Córrelo hoy, no el día del lanzamiento.

Esto no reemplaza a un auditor humano ni a las herramientas que ya usas. Razona sobre tu código como lo haría un investigador de seguridad, y eso ya es más de lo que tuvo la mayoría de las apps que hay. La app que nadie revisó es la que alguien más va a revisar por ti.

MÁS GUÍAS GRATIS EN [377CREATIVE.COM/RECURSOS](https://377creative.com/recursos)