

Agente PagoKit · Stripe, Mercado Pago y Lemon Squeezy en un comando

CATEGORÍA · CONECTORES, MCP E INTEGRACIÓN

Plugin gratis de Claude Code que escanea tu proyecto, hace 3 preguntas (país, una vez o suscripción, métodos locales) y elige Stripe, Mercado Pago, Wompi o Lemon Squeezy según tu caso. Genera ~14 archivos del checkout completo — botón, webhook firmado, portal del cliente, refunds y schema de DB — co

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

Plugin gratis de Claude Code que escanea tu proyecto, hace 3 preguntas (país, una vez o suscripción, métodos locales) y elige Stripe, Mercado Pago, Wompi o Lemon Squeezy según tu caso. Genera ~14 archivos del checkout completo — botón, webhook firmado, portal del cliente, refunds y schema de DB —

PARA QUÉ SIRVE

Plugin gratis de Claude Code que escanea tu proyecto, hace 3 preguntas (país, una vez o suscripción, métodos locales) y elige Stripe, Mercado Pago, Wompi o Lemon Squeezy según tu caso. Genera ~14 archivos del checkout completo — botón, webhook firmado, portal del cliente, refunds y schema de DB — con 5 candados deterministas que bloquean publicar API keys, webhooks sin firma o `.env`` fuera de `.gitignore``. Cubre Next.js App Router y Express en Fase 1, con roadmap pública a NestJS, FastAPI, Django, Laravel, Rails, Hono, SvelteKit, Astro y Go. Guía paso a paso explicada en lenguaje simple — qué es, cómo se instala, qué responder a cada pregunta, qué proveedor te conviene según tus clientes, qué hace cada uno de los 14 archivos generados, qué protege cada candado, y cuándo NO usarlo si tu stack está fuera de Fase 1.

DÓNDE APLICARLO

Nivel Avanzado. Temas: pagokit, pagos. Incluye el código de Hainrixz/agente-pagokit.

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

Plugin gratis de Claude Code que escanea tu proyecto, hace 3 preguntas (país, una vez o suscripción, métodos locales) y elige Stripe, Mercado Pago, Wompi o Lemon Squeezy según tu caso. Genera ~14 archivos del checkout completo — botón, webhook firmado, portal del cliente, refunds y schema de DB — con 5 candados deterministas que bloquean publicar API keys, webhooks sin firma o ``.env`` fuera de ``.gitignore``.

Cubre Next.js App Router y Express en Fase 1, con roadmap pública a NestJS, FastAPI, Django, Laravel, Rails, Hono, SvelteKit, Astro y Go. Guía paso a paso explicada en lenguaje simple — qué es, cómo se instala, qué responder a cada pregunta, qué proveedor te conviene según tus clientes, qué hace cada uno de los 14 archivos generados, qué protege cada candado, y cuándo NO usarlo si tu stack está fuera de Fase 1.

INSTALACIÓN

agente-pagokit

Plugin Claude Code que elige proveedor y escribe checkout, webhook firmado y refunds en tu proyecto.

EL CÓDIGO ESTÁ AQUÍ

<https://github.com/Hainrixz/agente-pagokit>
Licencia MIT · JavaScript

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/Hainrixz/agente-pagokit>

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/Hainrixz/agente-pagokit.git
cd agente-pagokit

# 2. Instala las dependencias (Node)
npm install

# 3. Instalalo como plugin (dentro de Claude Code)
/plugin marketplace add Hainrixz/agente-pagokit

# 4. Comprueba la lista de plugins
/plugin
```

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

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: Conectores, MCP e integración

Nivel: Avanzado

Lenguaje: Go

Temas: pagokit, pagos

Código: <https://github.com/Hainrixz/agente-pagokit>

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)