GUÍA NODESTUDIO

OpenReply

Comentario a DM automático en Instagram, con verificación real de seguidor, corriendo en tu propia infraestructura y sin pagar suscripción.

⏱ 90 minutos

1 Levantalo con Claude Code

OpenReply es el ManyChat open source: alguien comenta una palabra clave en tu reel y le llega un DM con tu recurso. No hace falta que sepas de terminal. Abrí Claude Code en una carpeta vacía y pegale esto — el link del repo va adentro, por si el prompt viaja solo:

REPO github.com/diwenne/openreply

PROMPT PARA PEGAR EN CLAUDE CODE

```
Quiero autohospedar OpenReply, el clon open source de ManyChat para Instagram:
https://github.com/diwenne/openreply

Forkealo a mi cuenta de GitHub y desplegá la web en Vercel. Para los datos usá
Supabase (Postgres, connection string en modo Session, puerto 5432) y Redis Cloud
Essentials en su plan gratis de 30 MB — la cola es BullMQ y necesita Redis nativo
por TCP, un Redis por HTTP no sirve. Para los magic links usá Resend.

Generá los secretos de .env.example: ENCRYPTION_KEY tiene que ser hex de 64
caracteres exactos. Poné mi mail en ALLOWED_EMAILS y dejame todas las variables
listas para pegar en Vercel de una sola vez. Decime qué cuentas tengo que crear
yo.
```

ELEGÍ BIEN LOS PLANES GRATIS

El worker vive prendido 24/7, así que cualquier free tier medido *por tiempo* o *por cantidad de comandos* se agota solo. Upstash (500k comandos/mes) y Neon (100 CU-hours) se consumen enteros sin que mandes un solo DM. Redis Cloud y Supabase limitan por memoria, que es lo que te conviene.

DÓNDE CORRE CADA COSA

La web y el webhook van en Vercel. El **worker** no: consume la cola y tiene que estar siempre encendido, cosa que una función serverless no puede hacer. Cualquier máquina que quede prendida sirve — una laptop vieja, una Raspberry, una VM gratis. Sale solo hacia afuera, no hay que abrirle ningún puerto.

2 Conectalo con Instagram

Esta es la parte que más gente abandona. Va en este orden y no en otro, porque la app de Meta necesita la URL que recién existe después del deploy.

Creá la app de Meta

En developers.facebook.com, producto **Instagram** → API con inicio de sesión de Instagram. Necesitás cuatro permisos: `instagram_business_basic`, `manage_messages`, `manage_comments` y `manage_insights`.

> **LOS DOS CAMPOS QUE SE CONFUNDEN SIEMPRE**
>
> `/api/webhook` es por donde **Meta te avisa** de comentarios y taps.
> `/api/instagram/callback` es por donde **vos volvés** tras dar permisos. Distinto sentido, distinto campo del panel.

Suscribí los campos correctos

En la tabla de campos del webhook prendé `comments`, `messages` y `messaging_postbacks`. El tercero es el que avisa cuando alguien toca un botón del DM: sin él, el follow gate recibe el tap y no se entera nunca.

Armá la campaña

Ya en el panel de OpenReply: elegís el reel, la palabra clave, el mensaje y el link. Si activás el **follow gate**, quien comenta recibe primero un DM con un botón; al tocarlo se consulta contra la API si te sigue de verdad, y recién ahí sale el link.

> **POR QUÉ EL GATE NECESITA ESE BOTÓN**
>
> Instagram solo te deja consultar si alguien te sigue *después* de que esa persona interactúe con vos. Un comentario no alcanza. El tap del botón es lo que habilita la consulta — por eso todas las herramientas del rubro tienen ese paso.

3 Problemas reales al probarlo

Todo esto pasó de verdad montando el sistema. No es troubleshooting genérico sacado del README.

NO LLEGA NINGÚN COMENTARIO, PERO EL WEBHOOK "ANDA"

Instagram **no entrega webhooks de `comments` mientras la app esté en modo Desarrollo**, aunque el campo figure suscrito a nivel app y a nivel cuenta. Lo tramposo: los eventos de `messages` sí llegan, así que parece que todo funciona.

Hay que pasar la app a **En vivo**. No requiere App Review — publicada o no, solo los testers pueden conectar su cuenta.

"CAN'T REACH DATABASE SERVER", PERO LA BASE ESTÁ PERFECTA

El pool de conexiones no traía tope, así que abría hasta 10 por proceso. Cada instancia serverless arma el suyo, y el pooler gratis de Supabase corta en 15: con dos instancias ya estaba lleno. Prisma lo reporta como si la base estuviera caída.

Se arregla pasándole un límite explícito al pool: 1 en la web, 3 en el worker.

EL MAGIC LINK NUNCA LLEGA

Resend, sin dominio propio verificado, **solo entrega al mail con el que creaste la cuenta de Resend**. Si tu usuario del panel es otra dirección, el envío se rechaza y la pantalla muestra un error de servidor genérico.

Que coincidan las tres: cuenta de Resend, `ALLOWED_EMAILS` y el mail que tipeás al entrar.

CAMBIASTE UNA VARIABLE Y NO PASÓ NADA

En Vercel, editar una variable de entorno **no afecta al deploy que está corriendo**. Hay que redeployar siempre. Si además venías con sesión abierta, limpiá las cookies del sitio: una sesión vieja apuntando a otra base genera un bucle de redirecciones que parece un bug de la app.

3 Problemas reales (continuación)

LE MANDÓ EL DM A GENTE QUE YA LO HABÍA RECIBIDO

El worker barre comentarios cada pocos minutos como red por si el webhook falla, y por defecto mira **72 horas hacia atrás** sin importar cuándo creaste la campaña. Si la creás sobre un reel que ya tiene comentarios, les escribe a todos.

Antes de armar campañas sobre posts viejos, bajá esa ventana con `COMMENT_POLL_LOOKBACK_HOURS` a 1 o 3.

DEJALO AVISANDO SOLO

OpenReply expone `/api/health`, que devuelve 200 si base, Redis, cola y worker están bien, y **503 si algo falla**. Apuntale un monitor gratis tipo UptimeRobot cada 5 minutos y listo.

Medido en el sistema real: desde que el worker muere hasta que llega el mail pasan unos 7 minutos. Mucho mejor que enterarte por un seguidor preguntando por qué no le llegó el recurso.

UN DATO PARA DECIDIR SI EL GATE TE CONVIENE

El follow gate suma seguidores pero cuesta entregas: hay gente que comenta, recibe el pedido de seguir y no completa el paso. Mirá esa proporción en tus propios números antes de prenderlo en todas las campañas.



NodeStudio

Automatizamos procesos de negocio con IA y sistemas a medida.

nodestudioai.com