

GUÍA PASO A PASO

Tu calendario de contenidos con aprobación de cliente

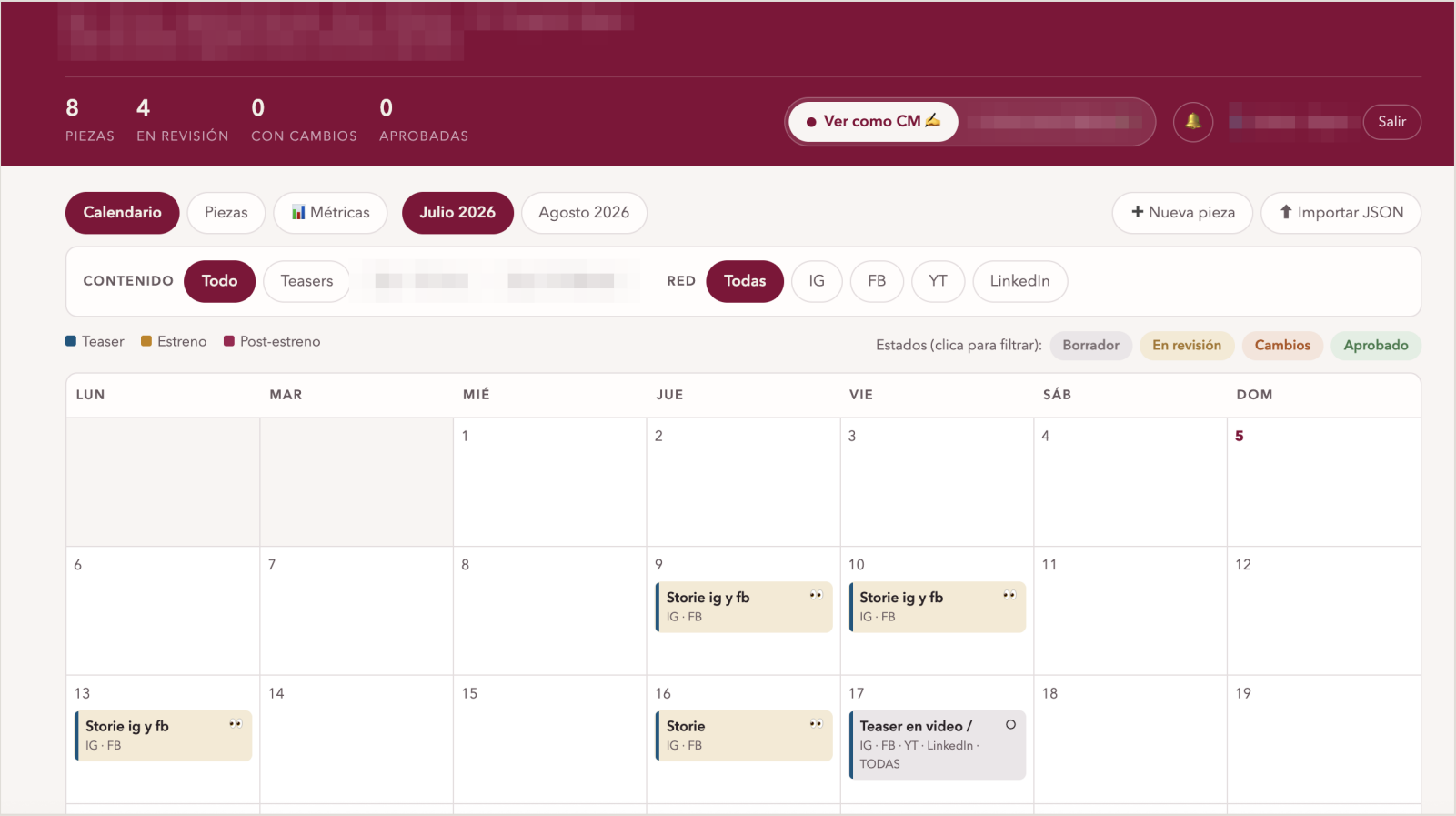
Monta tu propia app sin programar:
copiando un prompt.

Claude Code · Supabase · Resend

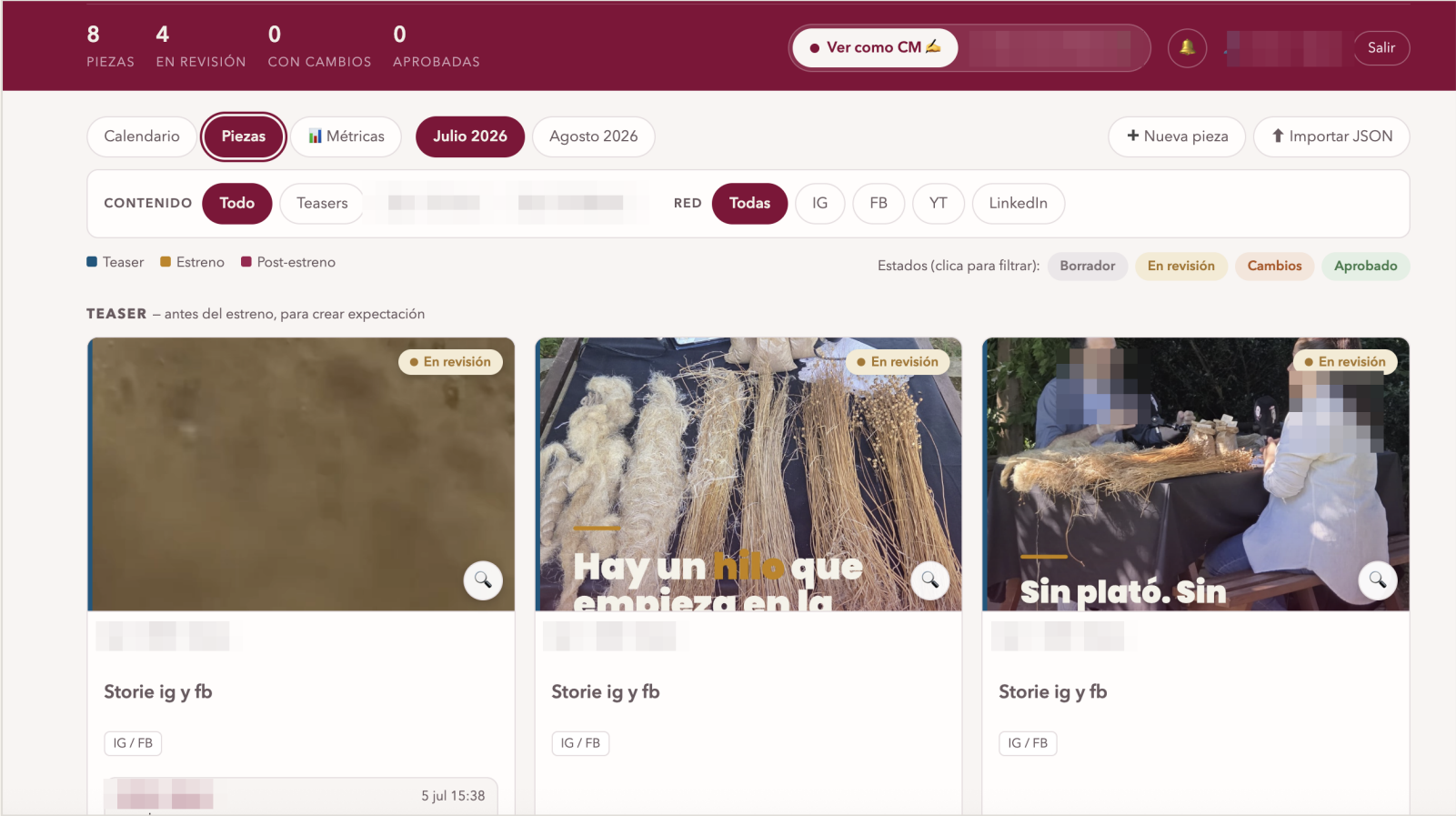


Qué vas a montar

Una app donde tu equipo sube las piezas, tu cliente las aprueba con un clic y todos reciben un email cuando algo pasa. Sin excels compartidos, sin cadenas de WhatsApp, sin "¿viste el post que te mandé?".



Vista calendario: arrastra una pieza a otro día y cambia de fecha.



Vista piezas: galería con estados, filtros y comentarios.

Todo vive en un único archivo HTML + una base de datos gratuita.

La montas en una tarde y la mantienes pidiéndole cambios a Claude en texto normal.

Qué necesitas antes de empezar

Claude Code (Anthropic)

El "programador": construye y mantiene la app por ti.

Desde ~20 €/mes

Supabase

Base de datos, login de usuarios y almacenamiento de fotos y vídeos.

Gratis (hasta 1 GB)

Resend

Los emails automáticos: "el cliente ha aprobado tu pieza".

Gratis (3.000/mes)

Hosting con tu dominio

Donde vive la app: un único archivo HTML.

El que ya tengas (~3-9 €/mes)

Un dominio propio

Para el subdominio (calendario.tudominio.com) y para enviar emails.

El que ya tengas (~10 €/año)

Tiempo

Una tarde tranquila.

~1-2 h la primera vez

Total si ya tienes dominio y hosting: el precio de Claude Code.

Todo lo demás, gratis.

Instrucciones: 5 pasos

- 1

Supabase

Crea cuenta en supabase.com, dale a "New project" y, en Settings API, copia la Project URL y la anon key. Mejor aún: conecta el MCP oficial de Supabase a Claude Code (`claude mcp add supabase`) y las tablas se crean solas.
- 2

Resend

Crea cuenta en resend.com, añade tu dominio en Domains, pega los 3 registros DNS donde gestionas tu dominio y, cuando esté "Verified", crea una API key (con "Sending access" basta).
- 3

Decide tu equipo

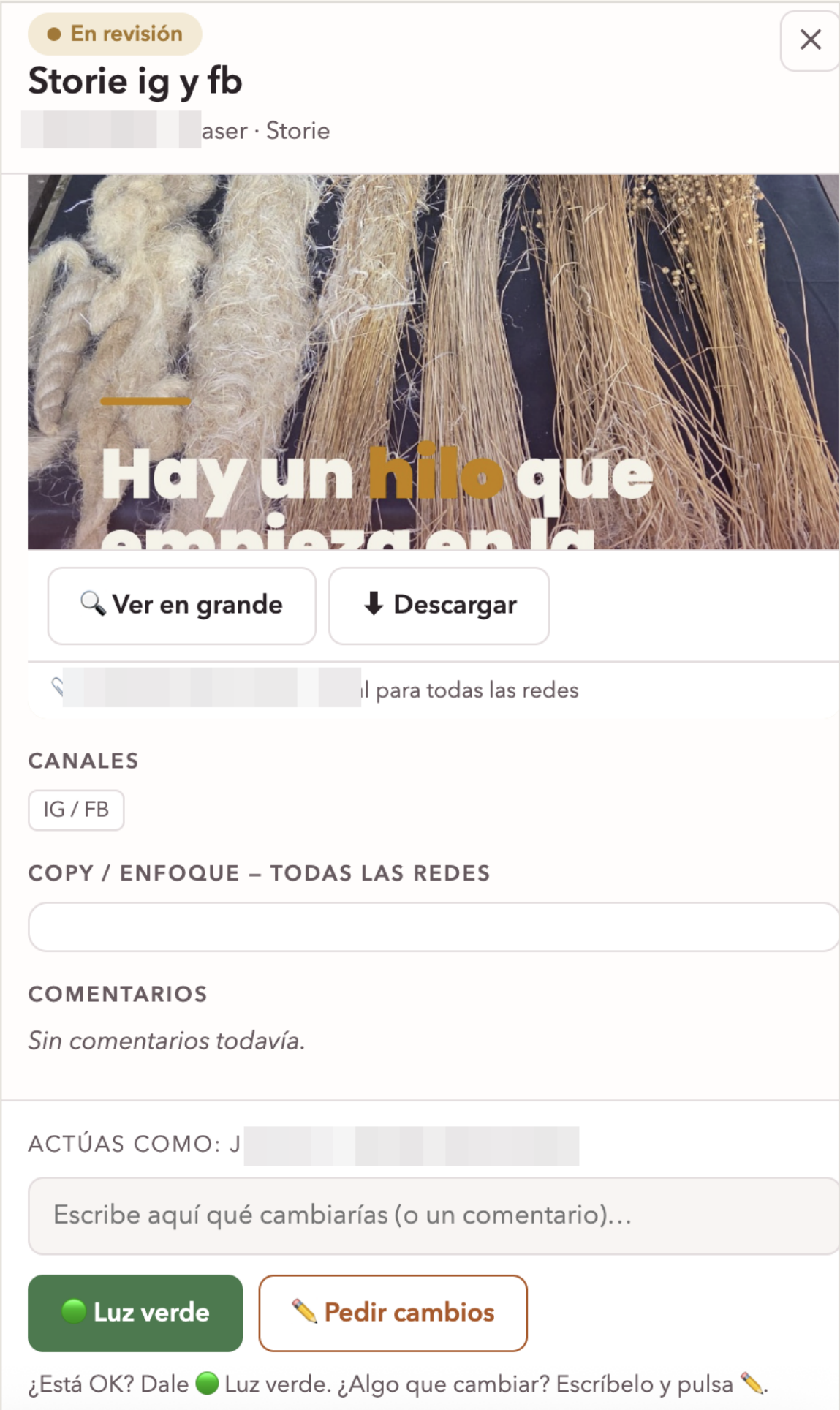
Quién sube contenido (tu equipo), quién aprueba (tu cliente), y sus emails. Rellena los [CORCHETES] del prompt con esos datos.
- 4

Pega el prompt en Claude Code

Está en las páginas 6–8. Pásale las credenciales cuando te las pida y pídele que verifique cada función en el navegador antes de dártela.
- 5

Sube el index.html a tu hosting

Apunta un subdominio a esa carpeta y reparte los accesos a tu equipo y a tu cliente.



Lo que ve tu cliente: dos botones y cero excusas.

Trucos de veterana

Pide que la app muestre su versión (v1, v2...) en el login

Cuando algo "no funcione", casi siempre es que tu hosting está sirviendo una versión vieja del archivo. Con el número de versión lo ves en un segundo, sin volverte loca.

Itera sin miedo

"El botón X no se entiende", "quiero que el cliente vea las fotos en grande"... Claude edita y te da el archivo nuevo. Cada mejora son minutos, no reuniones.

Prueba tú todos los roles antes de dárselo al cliente

Entra como admin, como equipo y como cliente. Lo que a ti te chirría, al cliente le frena. Que su primera impresión sea la buena.

Hacer las cosas difíciles no te hace más profesional, te hace más lento.

El prompt (1/3)

Cópialo entero (páginas 6–8), rellena los [CORCHETES] y pégalo en Claude Code. El texto es seleccionable.

Quiero que construyas una APP DE CALENDARIO DE CONTENIDOS con flujo de aprobación de cliente, lista para producción. Trabajo para [TU AGENCIA/MARCA] y el cliente es [NOMBRE DEL CLIENTE], para el proyecto [PROYECTO: p. ej. "lanzamiento de su podcast"].

STACK (no lo cambies):

- Un ÚNICO archivo index.html estático (HTML + CSS + JS vanilla, sin frameworks ni build steps) que se sube a un hosting normal. Vivirá en `https://[SUBDOMINIO.TUDOMINIO.COM]`.
- Supabase como backend: base de datos, autenticación y almacenamiento de fotos/vídeos. Cárgalo por CDN. Te daré la URL del proyecto y la anon key (o usa el MCP de Supabase).
- Resend para los emails automáticos, llamado desde una Edge Function de Supabase disparada por un trigger de base de datos. Te daré la API key. Remitente: `notificaciones@[TUDOMINIO.COM]`.
- Diseño: usa la identidad de [CLIENTE]: color principal [#HEX], secundarios [#HEX], tipografía [NOMBRE O "una sans limpia"]. Elegante y sobrio, no genérico.

USUARIOS Y ROLES (login con selector de usuario + contraseña, sesión persistente):

- [TU NOMBRE] (admin): todo, + conmutador "ver como" los demás roles.
- Equipo / community manager (editor): crea piezas, edita, sube material, envía a revisión.
- [NOMBRE CLIENTE] (revisor/a): comenta, aprueba o pide cambios. NO puede editar contenido.
- Viewer (solo lectura): puede mirarlo todo y descargar, tocar nada. Refuerza esto con políticas RLS en la base de datos, no solo ocultando botones.

Créalos como usuarios reales de Supabase Auth (emails con alias tipo `tucorreo+rol@gmail.com`).

El prompt (2/3)

FUNCIONALIDADES:

1. Vista CALENDARIO mensual y vista PIEZAS (galería de tarjetas con foto/vídeo grande, copy completo, comentarios y botones de acción inline). Filtros clicables por contenido, por red social (IG/FB/YT/LinkedIn) y por estado.
2. Cada pieza: título, fecha, formato, redes, copy, foto o vídeo (hasta 50 MB, calidad original, con botones "Ver en grande" (lightbox) y "Descargar"), estado y comentarios.
3. Versiones POR RED: una pieza puede tener copy y foto/vídeo distintos para cada red. Chips de red clicables que se agrupan si comparten versión ("IG / FB") y se separan solos cuando una red tiene su propia versión.
4. Flujo de estados: Borrador → En revisión → (cliente) Luz verde o Pedir cambios → Aprobado. Pedir cambios EXIGE escribir el motivo. El cliente puede cambiar de opinión en ambos sentidos. Guía visible que le explique al cliente qué hacer.
5. Crear piezas: botón "+ Nueva pieza", un "+" en cada día del calendario (fecha prellenada), edición de detalles a posteriori, e "Importar JSON" para volcar calendarios generados con IA (con anti-duplicados por fecha+título).
6. Drag & drop: arrastrar una pieza a otro día del calendario cambia su fecha.

El prompt (3/3)

7. TIEMPO REAL: los cambios de cualquier usuario aparecen al instante en los demás navegadores (Supabase Realtime).
8. CAMPANITA de notificaciones: historial de actividad con contador de no leídos por usuario; clicar una notificación abre la pieza.
9. EMAILS automáticos (trigger de BD → Edge Function → Resend, con plantilla en los colores del cliente): al cliente cuando algo pasa a revisión; a mi equipo cuando el cliente aprueba, pide cambios o comenta; a mí cuando el equipo sube material nuevo. Nunca notificar al autor de la acción. Destinatarios en una tabla de configuración.
10. Móvil impecable: el botón "atrás" cierra paneles en vez de salir de la web, formularios cómodos con teclado numérico, nada que se pise.
11. Muestra la VERSIÓN de la app (v1, v2...) en el login y súbela en cada iteración.
12. @MENCIONES en los comentarios: si un comentario contiene @nombre de cualquier usuario de la app (p. ej. @cliente, @admin), envíale un email automático con el texto del comentario citado y un botón que enlace a la app. Los comentarios sin mención no generan ningún email (solo aparecen en el historial de notificaciones interno). Resalta las menciones en negrita dentro de los comentarios, acepta el nombre con y sin tilde, nunca notifiques al propio autor, y pon una pista en el placeholder del campo de comentario (ej.: "Comenta... @nombre le avisa por email").

MÉTODO DE TRABAJO (importante):

- Ve por fases: primero base de datos y login, luego calendario y piezas, luego flujo de aprobación, luego notificaciones. Verifica CADA fase abriendo la app en un navegador y probando el flujo real (login, crear, subir foto, aprobar) antes de seguir.
- Seguridad: RLS activado en todas las tablas; escritura solo para los roles que corresponda; el bucket de archivos con lectura pública y escritura autenticada.
- Al terminar: dame el index.html final, las credenciales de los 4 usuarios, y una lista de qué probaste y cómo.

Extra: métricas en la misma app

El cliente ve alcance e interacciones sin salir de la app — y tú te ahorras la herramienta de informes. Solo necesitas un token de la API de Meta. Suena a cosa de programadores; son 10 minutos de clics.

- 1 El Instagram del cliente debe ser Business o Creator y estar vinculado a su página de Facebook. Sin eso, no hay API.
- 2 Entra en developers.facebook.com con tu Facebook, crea cuenta de desarrollador (gratis) y dale a "Crear app", tipo Negocio.
- 3 Dentro de la app, añade el producto de Instagram (Graph API).
- 4 Abre el Graph API Explorer (menú Herramientas), elige tu app y genera un token con permisos: `instagram_basic`, `instagram_manage_insights`, `pages_show_list`, `pages_read_engagement` y `read_insights`.
- 5 Ese token dura 1 hora: pásaselo a Claude tal cual, él lo cambia por uno de larga duración (60 días) y te guía si algo falla.
- 6 Mientras la app sea para tus propias cuentas (modo desarrollo), no necesitas pasar la revisión de Meta.

Añade esto al final del prompt:

EXTRA — MÉTRICAS:

- Añade una vista MÉTRICAS que lea de la Graph API de Meta (Instagram y Facebook): alcance, interacciones y seguidores por pieza publicada y por cuenta.
- Te daré un token de larga duración de Meta. Guárdalo en la tabla de configuración, editable desde el panel de admin (nunca hardcodeado en el HTML).
- Los tokens caducan a los ~60 días: si una llamada falla por token caducado, muestra un aviso claro en la vista de métricas con instrucciones para renovarlo, sin romper nada.
- Cachea las métricas en la base de datos y refréscalas como mucho cada pocas horas (hay un límite de ~200 llamadas/hora por cuenta).
- Guíame paso a paso para crear la app en developers.facebook.com y sacar el token.

Móntala esta tarde.

Y si prefieres que la tarde te la ahorre yo,
puedes ver todos mis servicios en mi web:

julietabolullo.com

¿Quieres contratar un servicio? Escríbeme.

WhatsApp

julieta@laagenciadeia.com