

Consejos y herramientas de IA #1 — octubre de 2026

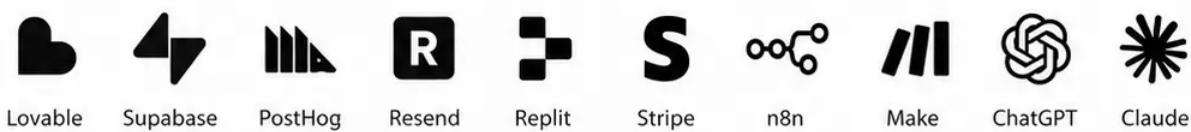
28 sept 2026 · 12 min de lectura · Por [Viktor Zaika](#)

Tres combinaciones prácticas de herramientas para probar una idea, crear un MVP y automatizar el trabajo, con prompts, diagramas y guías descargables.

- Herramientas de IA
- Gestión de productos
- MVP
- Automatización
- No-código

prompts&stacks

for your growth



Ser capaz de crear su propio proyecto ya no es la capacidad de los ingenieros solos. Los buenos gerentes de productos deben ser capaces de construir los PoCs, incluso MVPs y pruebas manuales y tener algo que les ayude a comunicar las ideas. No sólo eso, sino que prácticamente cualquier persona experta en tecnología hoy en día tiene que utilizar algunas tuberías de productividad personal que aumentan su rendimiento, darles más tiempo libre en un día, ayudarles a idear, aumentar su observabilidad en línea, etc. Así que aquí les voy a dar los consejos más esenciales, fáciles de leer y fáciles de usar, que les ayudarán inmensamente en estos escenarios:

 Lovable Página + interfaz de usuario	 Supabase Datos + Auth	 PostHog Acontecimientos + repetición	 Reenviar Correo electrónico
 Replit Aplicaciones de motor	 n8n Flujos de trabajo	 Marca Integraciones	 Raya Pagos

Las herramientas discutidas a continuación. Toque cualquier tarjeta para la documentación de ese producto o página de integración. El icono de Lovable es de Lovable; los otros son de Iconos simples.

1. Usted quiere probar la idea, no construir la empresa

Digamos que usted tiene una idea para algunos SaaS, mercado, herramienta de IA, lo que sea. El error aquí sería empezar inmediatamente a construir todo el asunto. No sabes si alguien lo quiere todavía.

Mi pila para esto sería:

Lovable + Supabase + PostHog + Reenviar

Lovable para la página y tal vez alguna interacción muy básica. Supabase para almacenar las inscripciones y cualquier pequeño motor que necesites. PostHog para entender realmente lo que la gente está haciendo en lugar de sólo mirar el contador de página. Reenviar para correos electrónicos de confirmación, listas de espera, invitaciones de acceso temprano, etc.

PostHog es particularmente útil aquí porque se puede ir de números de tráfico básicos a objetivos de conversión y embudos e incluso reproducciones de sesión, lo que significa que se puede ver literalmente donde la gente hace clic, donde se confunden y donde desaparecen.

Un giro realmente útil aquí es fingir la parte cara antes de construirlo.

Digamos que su idea es un servicio de IA que analiza un PDF y devuelve un informe complicado. Su landing page ya puede permitir que el usuario suba el PDF y pulse "Analyze". Detrás de las escenas se puede recibir el archivo, hacer las primeras 10 peticiones manualmente, y enviar el resultado por correo electrónico.

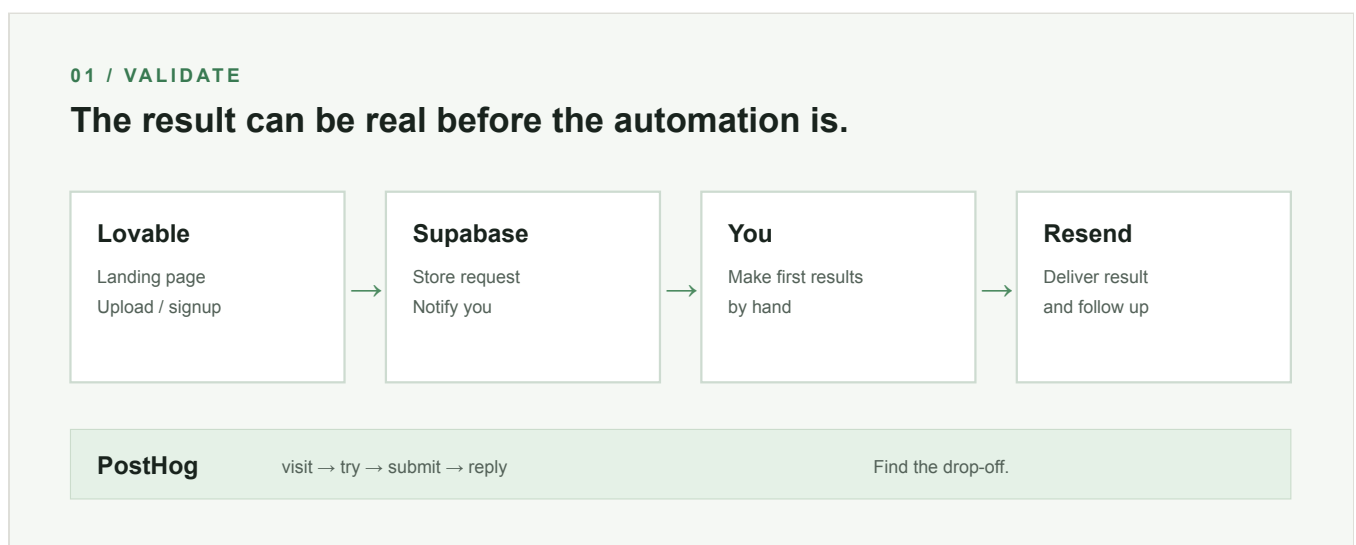
Sí, es una automatización falsa.

Pero usted está probando si la gente realmente quiere el resultado antes de pasar dos semanas construyendo el sistema que lo crea automáticamente.

Para este escenario en particular:

Página de aterrizaje → registro/carga → Supabase → notificación a usted → crear manualmente resultado → Reenviar correo electrónico

Una vez que la gente realmente comienza a usarlo, automatiza el medio.



Un posible flujo de trabajo de validación. Las pruebas de resultados hechas a mano exigen antes de que exista la costosa automatización. Integración de Supabase de Lovable, Embudos PostHog, y API de correo electrónico de Reenviar Soporta las piezas que se muestran aquí. Diagrama de Viktor Zaika. Pulsa la imagen para verla a tamaño completo.

Otra cosa útil: rastrear eventos, no sólo visitas.

No solo mida:

1.400 personas visitadas

Medida:

382 clicado intentarlo

117 comenzó el formulario

83 lo terminó.

19 respondió al correo electrónico

Eso te dice mucho más sobre si la idea es interesante. Si tienes eventos reales cuenta, mi [Calculadora de embudo de conversión](#) puede mostrar dónde ocurre la mayor caída antes de poner esos pasos en un embudo PostHog.

Valida tu idea

Copiar prompt ↗

Quiero validar esta idea de producto: [DESCRIBE IDEA].

No quiero construir el producto completo todavía.

Diseña la arquitectura de validación más pequeña posible usando Lovable, Supabase, PostHog y Reenviar.

Explique:

1. lo que realmente se debe construir
2. lo que puede ser falsificado manualmente detrás de las escenas
3. qué datos deben almacenarse en Supabase
4. que eventos PostHog debo rastrear
5. qué embudo de conversión debería crear
6. qué correos electrónicos se deben enviar a través de Reenviar
7. lo que contaría como evidencia suficiente para proceder a un MVP

Mantenga la arquitectura intencionalmente simple. No introduzca servicios adicionales a menos que haya una muy buena razón.

2. En realidad necesitas una aplicación de trabajo ahora

Una vez que usted necesita usuarios para iniciar sesión, guardar algo, volver mañana, tal vez pagar, tal vez subir archivos, entonces ya está construyendo una aplicación.

Aquí lo dividiría en dos caminos.

Caso A: SaaS normal o aplicación web

Lovable + Supabase + Stripe + Reenviar + PostHog

Esto es honestamente suficiente para un número sorprendentemente grande de MVPs. Si usted elige esta pila, [conectar un proyecto de Supabase que posee a Lovable](#); Lovable actualmente también ofrece su propio motor incorporado por defecto.

Supabase te da real Postgres debajo, más Seguridad de la Auth y del nivel de la fila, así que su interfaz generada no tiene que convertirse en su modelo de seguridad.

Y esta es una de esas cosas técnicas aburridas que las aplicaciones generadas por IA pueden arruinar horriblemente: la autorización no es lo mismo que ocultar un botón.

Si Alice entra, no debería ser capaz de cambiar alguna petición y de repente leer los registros de Bob.

Ahí es donde Supabase RLS realmente importa. Auth identifica al usuario, mientras que RLS decide a qué filas se le permite acceder al usuario. Guía RLS de Supabase explica cómo funcionan conjuntamente las subvenciones y las políticas de la base de datos.

Para las operaciones del servidor, llamadas API, Anzuelos de rayas o algún paso de generación de IA, Funciones de borde de supabase son una siguiente capa bastante conveniente. Se ejecuta el lado del servidor TypeScript y se puede integrar con Auth, Postgres y APIs externas.

Por lo tanto, un MVP muy normal podría parecer:

Interfaz lovable → Supabase Auth → Postgres con RLS → Función de borde → API externa

Entonces:

Stripe webhook → Función Edge → actualización de la suscripción en Postgres

Y:

acción de usuario → Función de borde → Reenviar

Eso ya es una arquitectura bastante real.

Caso B: el motor es el producto

Si estás construyendo algo con trabajadores, API inusuales, raspado, procesamiento de archivos, colas, paquetes Python o alguna lógica de backend rara, probablemente me movería a Replit en lugar de forzar todo a través de un constructor de frontend.

Replit Agent puede generar la aplicación a partir del lenguaje natural, mientras que Replit también proporciona despliegue e integración de bases de datos en el mismo entorno.

Por ejemplo:

Usted quiere construir una herramienta de monitoreo de la competencia.

Entrada:

URLs del competidor

Sistema:

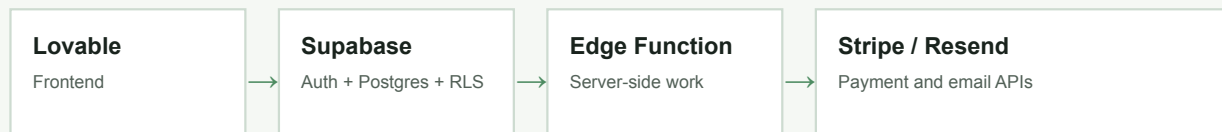
buscar páginas → comparar con la versión anterior → preguntar LLM qué cambió materialmente → guardar resultado → enviar notificación

Eso ya es mucho más natural una aplicación motor que una interfaz de usuario bastante generada.

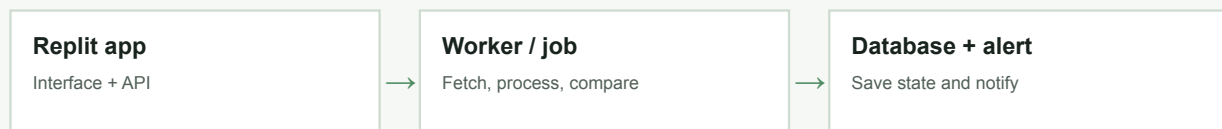
02 / BUILD

Choose the architecture by the hard part.

A / WEB SAAS



B / BACKEND-HEAVY APP



Dos rutas ilustrativas MVP, elegidas por lo que el producto realmente necesita. La conexión Supabase de Lovable y Funciones de borde; el camino de motor-pesado utiliza Agente de replit. Las flechas son ejemplos de arquitectura, no una afirmación de que estos productos se integran automáticamente. Diagrama de Viktor Zaika. Toque la imagen para verla a tamaño completo.

Una regla muy útil aquí:

No ponga sus claves de API secretas en la interfaz.

Tecla OpenAI, secreto de rayas, tecla Reenviar, lo que sea. Si el navegador puede verlo, asumir que alguien más puede verlo también.

Ponga estas operaciones detrás de un servidor endpoint o función. Guía de seguridad de datos de Supabase cubre este límite lado servidor.

Quiero construir este MVP: [DESCRIBE PRODUCT].

Ayúdame a elegir entre:

A. Lovable + Supabase

B. Replit

C. Una combinación de ambas

No elija en función de la herramienta que más le guste. Elija en función de los requisitos técnicos reales.

En primer lugar, identificar:

1. Requisitos de interfaz
2. autenticación
3. Entidades y relaciones de base de datos
4. Normas de autorización
5. API externos
6. trabajos de fondo o de larga duración
7. almacenamiento de archivos
8. pagos
9. correo electrónico transaccional
10. análisis

Entonces propone la arquitectura más pequeña que aún está técnicamente cuerda.

Dime explícitamente qué operaciones deben ocurrir del lado del servidor y qué secretos NUNCA deben ser expuestos en la interfaz.

Si usa Supabase, también proponga las tablas y las reglas RLS.

3. Usted quiere dejar de hacer algo manualmente cada maldito día

Esta podría ser la categoría más útil para la mayoría de la gente.

Usted no necesita necesariamente un producto.

Necesitas una tubería.

Mi defecto aquí probablemente sería:

n8n

Entonces conecta lo que ya uses alrededor.

Google Drive, Gmail, Slack, Telegram, Notion, Airtable, Supabase, OpenAI, Claude, API, etc.

Si quieres algo más fácil y muy visual, Marca es una alternativa muy razonable y apoya miles de integraciones, incluyendo Conexiones HTTP genéricas cuando el servicio que usted necesita no es soportado directamente.

Y la parte interesante comienza cuando dejas de pensar:

desencadenar → acción

y empezar a pensar:

entrada → entender → decidir → actuar → recordar

Por ejemplo, aquí hay un conducto personal realmente útil:

Gmail → n8n → LLM → clasificar → extracto de fecha límite → crear tarea → Notificación de Telegram

Pero no le pida al LLM que simplemente "lea mi correo electrónico y decida todo".

Hacer que devuelva datos estructurados:

```
{
  "important": true,
  "category": "invoice",
  "deadline": "2026-10-14",
  "action_required": "pay invoice",
  "confidence": 0.94
}
```

Copy

Ahora el flujo de trabajo puede realmente utilizar la salida de forma fiable. El analizador de salida estructurado de n8n es una forma de imponer la forma de un resultado.

Ese es un pequeño detalle técnico que hace que las automatizaciones de IA sean mucho menos frágiles.

Otra:

No permita que la IA realice acciones irreversibles directamente cuando pueda evitarlo.

Por ejemplo:

email → AI piensa que esto es spam → DELETE

Mal.

Mejor:

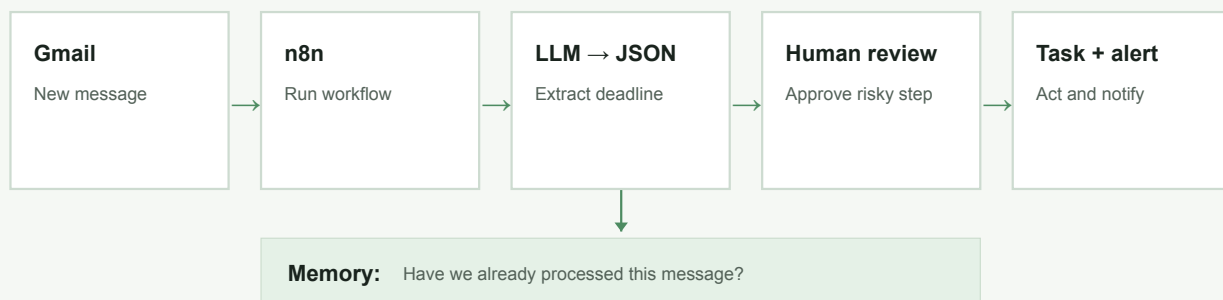
email → AI piensa que esto es spam → etiqueta "probablemente spam" → usted confirma

Lo mismo con enviar correos electrónicos, publicar contenido, eliminar archivos, actualizar los datos de producción, etc.

Ponga un fase de aprobación humana donde el costo de una decisión equivocada es alto.

03 / AUTOMATE

Input → understand → decide → act → remember



Un trabajo diario ilustrativo. Salida estructurada permite que los pasos posteriores utilicen campos; revisión humana mantiene las acciones de alto costo bajo su control. Diagrama de Viktor Zaika. Toque la imagen para verla a tamaño completo.

Algunas tuberías realmente útiles

Gasoducto de investigación personal

RSS / sitios web / boletines de noticias → n8n → extracto de texto → deduplicar → LLM clasifica relevancia → guardar interesantes → un digest de Telegram

En vez de leer 80 cosas, usted lee 7.

Reunión de la tubería

Calendario final de reuniones → transcripción aparece → LLM extrae decisiones y acciones → crear tareas → enviar un resumen breve

Conducto de contenido

Idea en Telegram → n8n → guardar en Notion → LLM lo categoriza → genera preguntas de investigación → espera

Observe que no estoy diciendo "generar el puesto".

La parte útil podría ser simplemente eliminar el estúpido trabajo de organización alrededor de escribirlo.

Observabilidad en línea

Google Alerts / Reddit search / X o API externa / comentarios de productos → flujo de trabajo → eliminar duplicados → clasificar sentimiento/tema → notificar sólo cuando algo interesante sucede

Así que en vez de googlearse, compañía o competidor cada pocos días, el sistema hace esa parte.

4. La parte genuinamente interesante: dejar que la IA produzca datos, no prosa

Esta es probablemente una de las mayores diferencias entre una automatización divertida y algo de lo que realmente puedes depender.

No preguntes:

Lee este ticket de apoyo y dime qué hacer.

Preguntar:

```
Analyze this support ticket and return ONLY valid JSON using this schema:
category: billing | bug | feature_request | account | other
urgency: 1 to 5
requires_human: boolean
customer_sentiment: positive | neutral | negative
summary: maximum 200 characters
```

Copy

Ahora otra parte de tu oleoducto puede en realidad enrutarlo.

urgencia > 4 → Slack

facturación → cola de facturación

requires_human = false → generar borrador

Los LLMs son mucho más útiles dentro del software cuando son tratados como una función poco fiable dentro de un sistema determinista más grande, en lugar de un tipo mágico que controla todo el asunto.

04 / DECIDE

Give each part of the job to the right kind of logic.

Normal code

```
price > 1000
```

Exact comparisons, IDs, math, access
control, deduplication

LLM + schema

```
{ urgency: 4, category: "billing" }
```

Messy text → structured output
Human approves expensive actions

Utilice código normal para reglas exactas y un LLM para entradas borrosas. El lado JSON ilustra el tipo de campos un n8n paso de salida estructurada Diagrama de Viktor Zaika. Pulsa la imagen para verla a tamaño completo.

5. Y dar su memoria de flujo de trabajo

Digamos que usted crea una tubería de investigación de IA.

Sin memoria:

El lunes encuentra el artículo X.

El martes vuelve a encontrar el artículo X.

El miércoles encuentra a alguien discutiendo el artículo X y orgullosamente te lo dice de nuevo.

Almacene lo que ya ha procesado.

Esto puede ser literalmente una tabla de Supabase:

```
source_url
hash
first_seen
summary
topic
```

Copy

Luego, antes de procesar algo caro:

¿Ya he visto esto?

Si es así, sáltalo.

Muy simple. De repente su tubería se siente unas 5 veces más inteligente.

6. No hacer automáticamente todo AI

Esto suena estúpido en un post llamado IA Tips&Tools, pero importa.

Si esto funciona:

```
price > 1000
```

Copy

no lo sustituya por:

```
Ask GPT whether this price appears relatively expensive in the context of the transact
```

Copy

Por favor.

El código normal es más rápido, más barato y determinista.

Utilice el LLM donde la entrada es borrosa.

Entendiendo el lenguaje.

Clasificación donde las reglas se vuelven horribles.

Resumiendo.

Extracción de documentos desordenados.

Coincidiendo con las cosas por significado.

Generando algo.

Deja las matemáticas, identificaciones, comparaciones exactas, control de acceso y la mayoría de las aburridas reglas de negocio a un código normal aburrido.

Aburrido es fantástico cuando funciona cada vez.

Las tres pilas que realmente recordaría

Comprobar si a alguien le importa

Lovable + Supabase + PostHog + Reenviar

Construir algo que la gente pueda usar

Lovable + Supabase para la web normal SaaS

o

Replit cuando la lógica de backend se convierte en la parte interesante

Luego añade Stripe, Reenviar y PostHog cuando realmente los necesites.

Automatice su propio trabajo

n8n + las herramientas que ya utiliza + LLM + una pequeña base de datos para la memoria

Y probablemente el mensaje más útil de todo este post:

Automatice su trabajo

Copiar prompt ↗

Actualmente hago esto manualmente:

[DESCRIBE THE PROCESS IN NORMAL HUMAN LANGUAGE]

Diseña una automatización para ello.

Separar el flujo de trabajo en:

pasos deterministas que deberían utilizar la lógica normal,

pasos borrosos donde un LLM realmente tiene sentido,

acciones que requieran la aprobación humana,

state que necesita ser almacenado para que el flujo de trabajo recuerde lo que ya hizo.

Prefiere n8n y servicios que ya uso. Mantenga el número de piezas móviles lo más pequeño posible.

Para cada paso LLM, defina la salida JSON estructurada exacta que debe devolver.

También dime cómo este flujo de trabajo puede fallar y lo que debería registrar para poder depurarlo más tarde.

Porque este es un poco el punto. IA no significa que de repente tienes que convertirte en un ingeniero. Pero tampoco significa que la ingeniería dejó de importar.

Ahora puedes construir una cantidad absurda de cosas sin saber todo lo que hay debajo de ella. Sólo necesitas saber lo suficiente para entender dónde están las partes peligrosas, dónde la IA es útil, y dónde realmente no debes dejar que sea estilo libre.