GUÍA NODESTUDIO

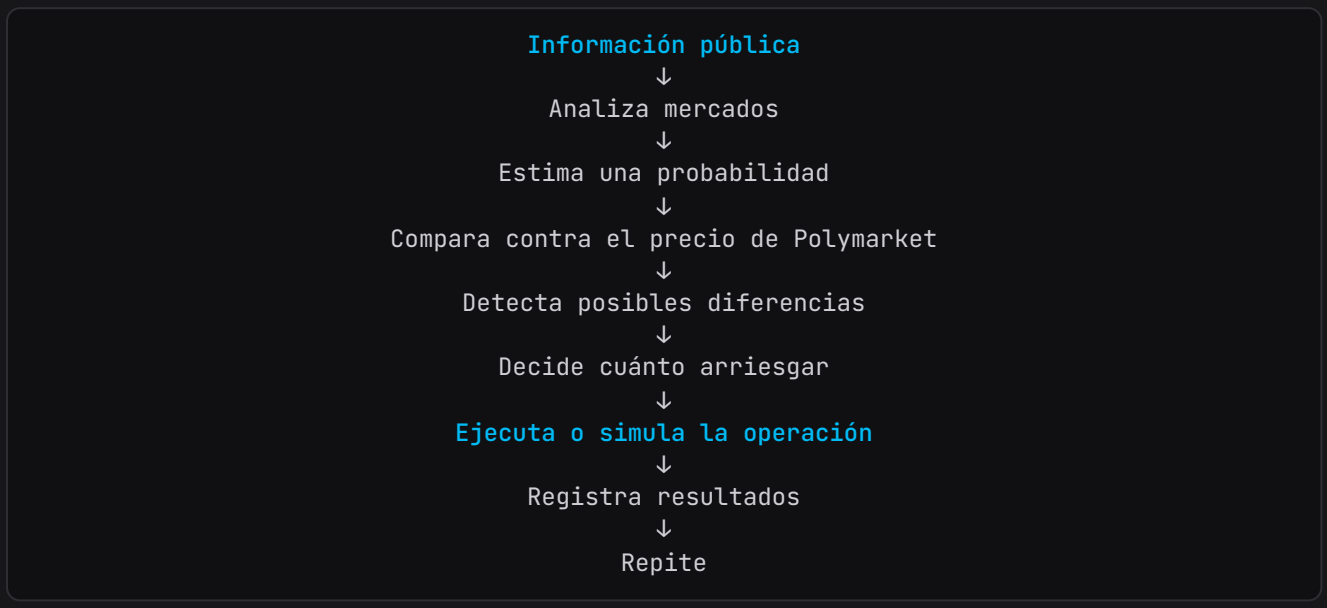
Cómo recrear el agente de Polymarket que intenta pagarse solo

Armás un agente que analiza mercados, estima probabilidades y registra cada decisión — empezando en simulación, sin plata real.

⏱ ~45 min para tenerlo corriendo en simulación

1 Qué vamos a construir

Un agente que corre solo en la nube, mira los mercados de Polymarket, se arma una opinión propia sobre qué probabilidad tiene cada cosa de pasar, y la compara contra el precio. Si el precio y su estimación difieren mucho, anota la oportunidad y dimensiona cuánto arriesgaría. Al principio no ejecuta nada: simula y registra.



De dónde sale la idea

El disparador es una publicación de Argona (`@Argona0x`): le dio 50 dólares a un Grok Bot con una sola instrucción — *"pagate solo o morís"* — y a las 48 horas afirmó que el bot tenía más de 5.000 dólares operando en Polymarket.

> **LEÉ ESTO ANTES DE SEGUIR**
>
> Ese resultado de $50 → ~$5.273 en 48 horas es **una afirmación del creador**. Esto es lo que afirma el creador; no pude verificarlo independientemente: no hay wallet pública, ni historial on-chain, ni capturas auditables. La cobertura que salió después repite el posteo sin verificarlo, y ya aparecieron cuentas copiando el mismo texto con otras cifras.
>
> Tratalo como lo que es: una idea interesante de arquitectura, no un rendimiento esperable. 48 horas no son una muestra. Un agente así puede perder todo el capital.

Qué te llevás igual

La parte reproducible no es el número: es el sistema. Un agente con computadora propia, una rutina que corre sola, un criterio de decisión explícito, límites de riesgo que no puede cambiar, y un registro de cada operación para poder medirse. Eso lo armás hoy, y sirve incluso si nunca le ponés un peso.

2 Qué necesitás

Obligatorio

- **Un agente con computadora en la nube.** Grok Bot cumple: según la documentación de xAI, cada cuenta tiene una computadora persistente en la nube con navegador, sistema de archivos y terminal, compartida entre tus bots.
- **Una suscripción paga de Grok.** Grok Bot no es gratis. No pude confirmar el precio exacto ni qué plan te alcanza: chequealo en la página de planes de tu cuenta antes de pagar.
- **Nada de plata al principio.** Para la fase de simulación no necesitás wallet, ni USDC, ni cuenta en Polymarket. Los datos de mercado son públicos.

Opcional (recién en fase 2)

- Cuenta de Polymarket con fondos, si algún día pasás a dinero real.
- Un bot de Telegram para recibir los resúmenes (página 12).

Alternativas si no tenés Grok Bot

La idea no depende de Grok. Necesitás tres cosas: que el agente corra sin tu máquina prendida, que pueda leer la web, y que pueda guardar archivos entre corridas.

Opción	Cómo cubrís lo que falta
Claude Code en tu compu	Corre local, así que la máquina tiene que estar prendida. Programás la rutina con `cron` o el Programador de tareas.
Claude / ChatGPT con tareas programadas	Corren en la nube, pero sin terminal propia. Sirve para investigar y avisarte; el registro lo llevás vos en una planilla.
Un VPS de 5 USD + un agente por API	Es la versión "a mano": más laburo, pero control total y el que sale más barato si lo dejás corriendo.

> **ANTES DE EMPEZAR**
>
> Polymarket bloquea 39 países por completo — entre ellos Estados Unidos, Brasil y buena parte de Europa — y algunas provincias de Canadá. Uruguay no aparece en esa lista al día de hoy, pero verificá el Help Center desde tu país. Usar VPN para saltear el bloqueo va contra los términos de uso: si lo hacés, te podés quedar sin los fondos.

3 Crear el agente y darle sus accesos

En Grok Bot creás un bot, le describís el trabajo, y le vas dando accesos a medida que los pide. No hay que armar ningún flujo con cajitas. Estas son las cinco capacidades que el sistema necesita y cómo se resuelve cada una:

Capacidad	Cómo la tiene	Estado
Navegador	El navegador de la computadora en la nube. Las sesiones quedan abiertas y se comparten entre tus bots.	Documentado
Terminal	Línea de comandos en la misma computadora. Ahí corre los scripts y las consultas a la API.	Documentado
Almacenamiento persistente	La carpeta `/workspace` guarda los archivos del proyecto y sobrevive las actualizaciones de la máquina.	Documentado
Ejecución programada	Las *routines* corren una skill por horario o por evento. Hasta 50 routines por bot.	Documentado
Notificaciones	Solo notificación del sistema operativo o push al celular. Telegram no viene nativo: lo armás vos (página 12).	Documentado

LO QUE SÍ SE CONFIRMA DE LA HISTORIA

Que el agente siga trabajando con tu laptop apagada es real y está en la documentación: cerrar la app, la laptop o el teléfono no detiene una rutina en curso. Y es cierto que no hace falta montar un VPS ni configurar claves de API para leer datos: la computadora ya viene con navegador y terminal, y los datos de mercado de Polymarket son públicos.

OJO CON ESTO

La computadora en la nube es **una sola por cuenta** y la comparten todos tus bots. Cualquier credencial o archivo que dejes ahí queda al alcance de los demás. Si vas a manejar algo sensible, no lo mezcles con otros bots.

Además, un bot corre **una sola tarea de computer-use a la vez**. No le encimes rutinas que peleen por la pantalla.

DOCUMENTACIÓN OFICIAL

Grok Bot — docs.x.ai/grok-bot/overview

Polymarket — docs.polymarket.com

4 Enseñarle a usar Polymarket

El creador cuenta que hizo una operación a mano mientras el agente miraba, y después convirtió ese flujo en una skill. Eso existe y tiene nombre: **"Teach a task"**. Abrís la conversación con vista de la computadora, elegís "Teach a task", describís el resultado que vas a mostrar, y hacés el flujo una vez. Graba hasta 10 minutos de interacción visible del navegador — no graba audio. Lo que sale es un **borrador** de skill: le faltan las reglas de decisión y los límites, que los agregás vos.

También podés hacerlo sin demostración: corré el proceso conversando y después pedile *"guardá el proceso que acabamos de usar como una skill llamada Análisis de mercados"*.

Qué automatizar y qué no

Parte del flujo	¿Se automatiza?
Leer mercados, precios y libros de órdenes	**Sí.** Es API pública, sin autenticación. Ni siquiera necesita el navegador.
Buscar contexto y mirar X	**Sí.** Navegador.
Estimar probabilidad, calcular edge y tamaño	**Sí.** Es la parte donde el agente aporta algo.
Registrar todo en un archivo	**Sí.** Un CSV en `/workspace`.
Login, contraseñas, 2FA, captchas	**No.** La documentación es explícita: esos pasos los hacés vos tomando el control. El bot pausa y te los pasa.
Ejecutar órdenes con dinero real	**No al principio.** Y cuando lo hagas, con aprobación previa por operación.

> **SI ALGÚN DÍA PASÁS A PLATA REAL**
>
> No le des tu clave privada. Polymarket tiene **session keys**: un firmante secundario que autorizás para operar por vos y que **no puede retirar fondos**. Lo limitás por ámbito (solo el libro de órdenes, por ejemplo) y lo revocás cuando quieras con `revokeSessionKey()`, que además cancela sus órdenes abiertas.
>
> Dos límites importantes: la autorización dura **180 días fijos** — no se puede configurar más corta — y **no tiene límite de gasto**. El techo de riesgo lo pone tu lógica, no la llave. Por eso el capital que dejás en esa wallet es tu verdadero límite de pérdida.

La forma segura de empezar es la de la página 11: el agente calcula todo y escribe la operación en un registro, pero no manda nada. Vos leés el resumen y decidís.

5 La estrategia de análisis

El creador dice que su agente corre cada ~10 minutos y revisa entre 500 y 1.000 mercados. Se puede: los endpoints de lectura de Polymarket son públicos y no piden autenticación.

1. Traer los mercados abiertos

```
# Gamma API — descubrimiento, sin auth
GET https://gamma-api.polymarket.com/markets/keyset?closed=false&limit=100
GET https://gamma-api.polymarket.com/public-search?q=elecciones
```

De cada mercado te importa la `question`, el `slug` y `clobTokenIds` — los IDs de los tokens YES y NO, que son la llave para pedir precios.

2. Traer precios y libro de órdenes

```
# CLOB API — lectura, sin auth. Hasta 500 tokens por request
POST https://clob.polymarket.com/prices
     [{"token_id":"<TOKEN_ID>","side":"BUY"}]

GET  https://clob.polymarket.com/book?token_id=<TOKEN_ID>
GET  https://clob.polymarket.com/spread?token_id=<TOKEN_ID>
```

3. Descartar sin piedad

Acá se gana la mayor parte. De 1.000 mercados, casi todos se tiran a la basura:

- **Spread ancho.** Si comprar y vender difiere mucho, la diferencia que creés tener ya se la come la entrada.
- **Libro flaco.** Mirá los niveles del `book`: si no hay tamaño suficiente al precio que querés, no podés entrar ni salir. Revisá también `minOrderSize` y `tickSize`.
- **Sin información pública real.** Si no hay fuentes para estimar nada, el agente no tiene ventaja: tiene una opinión.
- **Resolución muy lejana.** Plata inmovilizada meses y feedback que nunca llega.

4. Leer el precio como probabilidad

> **LA CONVERSIÓN**
> Si una acción YES cuesta **$0,40**, el mercado está reflejando aproximadamente un **40%** de probabilidad. Paga $1 si pasa, $0 si no. Esa lectura ignora fricciones: el spread, las comisiones y el costo de tener la plata ahí quieta.

5. Estimar por su cuenta

El agente busca información reciente, junta evidencia a favor y en contra, y recién ahí tira un número. El orden importa: primero la evidencia, después la probabilidad. Al revés, lo único que hace es justificar la primera intuición.

6 La regla del 8%

El filtro del creador: operar solo cuando su estimación difiere más de ~8% del precio del mercado. Se mide en **puntos porcentuales**, que es una resta, no una división.

```
Mercado (precio YES $0,42)  →  42%
Estimación del agente       →  53%
                               ──────
Diferencia                  →  11 puntos porcentuales  ✓ pasa el filtro del 8%
```

Si el mercado estuviera en 42% y el agente estimara 48%, son 6 pp: no pasa, y se descarta sin discusión. El punto del umbral no es encontrar más oportunidades, es tener menos.

> **LO MÁS IMPORTANTE DE ESTA PÁGINA**
>
> Que el modelo estime 53% **no significa que la probabilidad real sea 53%**. Es la opinión de un modelo con información incompleta. Del otro lado del precio hay gente con plata propia arriesgada, y a veces saben algo que el agente no.
>
> Un edge grande suele ser una señal de alarma, no un regalo. Si el agente ve 30 puntos de diferencia, lo más probable es que haya entendido mal la pregunta del mercado o las condiciones de resolución.

Las comisiones se comen parte del edge

Polymarket cobra solo al *taker* — el que toma el precio del libro. La fórmula oficial es `fee = shares × feeRate × p × (1 - p)`, así que la comisión es máxima justo en el 50% y baja hacia los extremos. La tasa depende de la categoría:

Categoría	feeRate	Costo en un mercado a ~$0,42
Geopolítica	0	Sin comisión
Finanzas, política, tech	0,04	~1,0 ¢ por share ≈ 1 pp de edge
Deportes, economía, cultura, clima	0,05	~1,2 ¢ por share ≈ 1,2 pp
Crypto	0,07	~1,7 ¢ por share ≈ 1,7 pp

Traducido: un edge de 11 pp en un mercado de crypto es más cerca de 9 pp reales, y eso antes de contar el spread. Por eso en el prompt de la página 9 el umbral se mide **neto de comisión estimada**, no sobre el número lindo.

Ser *maker* (dejar la orden en el libro y esperar) no paga comisión, pero te expone a que el precio se te vaya. Es un trade-off, no una optimización gratis.

7 Cuánto arriesgar: Kelly y el techo del 6%

Detectar una diferencia es la mitad. La otra mitad es cuánto poner. Apostar siempre lo mismo ignora que algunas oportunidades son mejores que otras; apostar de más te funde aunque tengas razón en promedio.

Kelly en una línea

El Kelly Criterion te dice qué fracción del capital maximiza el crecimiento a largo plazo. Para un mercado binario donde comprás YES a un precio `c` y tu probabilidad estimada es `p`:

```
fracción = (p - c) / (1 - c)

Ejemplo:  p = 0,53   c = 0,42
fracción = (0,53 - 0,42) / (1 - 0,42) = 0,11 / 0,58 = 19% del bankroll
```

Diecinueve por ciento en una sola apuesta. Y eso es lo que Kelly dice cuando tu probabilidad es correcta. Ese es justo el problema.

> **POR QUÉ NADIE USA KELLY COMPLETO**
>
> Kelly asume que tu `p` es la probabilidad verdadera. Acá `p` lo estimó un modelo de lenguaje leyendo noticias. Si el modelo está sobreconfiado — y los modelos son sobreconfiados — Kelly completo amplifica el error hasta fundirte.
>
> La práctica habitual es **Kelly fraccional**: usar un cuarto o la mitad del número. Con un cuarto, el 19% del ejemplo baja a **4,7%**.

El techo del 6%

El sistema original además limitaba cada posición a ~6% del bankroll. Los dos controles hacen cosas distintas y conviene tener los dos: Kelly fraccional escala con la calidad de la oportunidad, el techo del 6% es un freno duro para cuando el cálculo se va al carajo.

```
fracción_final = min( 0,25 × kelly , 0,06 )
```

Ni 1/4 ni 6% son números óptimos universales. Son valores conservadores y defendibles; el del 6% es simplemente el que usó el sistema del que habla la historia. Lo importante es que estén escritos, que el agente no los pueda tocar, y que apliquen sobre el bankroll de *hoy* — no sobre el que tenías cuando arrancaste.

8 El prompt principal (1 de 2)

Copiá las dos partes juntas, una abajo de la otra, y pegalas en el agente. Esto es la skill: la lógica vive acá, no en la cabeza del modelo.

PROMPT PRINCIPAL — PARTE 1

ROL

Sos un analista de mercados de predicción. Tu trabajo NO es operar mucho: es encontrar los pocos casos donde el precio de un mercado está claramente mal, y descartar todo lo demás. Descartar es el resultado esperado. Una corrida sin oportunidades es una corrida exitosa.

DATOS (endpoints públicos, sin autenticación)

1. Mercados abiertos:
GET https://gamma-api.polymarket.com/markets/keyset?closed=false&limit=100
Paginá con after_cursor. De cada mercado guardá: question, slug, categoria, clobTokenIds (tokens YES/NO), fecha de resolución.

2. Precios (hasta 500 por request):
POST https://clob.polymarket.com/prices con [{"token_id":"...","side":"BUY"}]

3. Profundidad y spread, solo para los candidatos que sobrevivan al filtro:
GET https://clob.polymarket.com/book?token_id=...
GET https://clob.polymarket.com/spread?token_id=...

FILTROS DUROS — aplicalos ANTES de investigar nada

Descartá el mercado, sin excepciones, si:

- el spread es mayor a 3 centavos;
- no hay al menos 10x el tamaño de tu posición disponible en los primeros niveles del libro;
- el precio está por debajo de $0,05 o por encima de $0,95 (ahí la comisión y el redondeo pesan más que cualquier edge que creas ver);
- resuelve a más de 90 días;
- no encontrás al menos 2 fuentes independientes y con fecha para estimar;
- no entendés con total precisión la condición de resolución. Si la regla es ambigua, se descarta. No la interpretes a tu favor.

INVESTIGACIÓN (solo para los que pasaron)

Para cada candidato, en este orden:

a. Leé la condición de resolución completa y escribí en una línea qué tiene que pasar exactamente para que YES pague.

b. Buscá información pública reciente. Priorizá: fuentes primarias (documentos oficiales, comunicados, datos publicados) > medios establecidos > agregadores. Anotá la FECHA de cada fuente.

c. Revisá sentimiento e información en X sobre el tema. Tratá X como una pista de qué se está discutiendo, NO como evidencia. Un posteo viral no es un dato.

d. Escribí explícitamente 2 razones a favor y 2 razones en contra. Si no podés armar las de contra, no entendiste el mercado: descartalo.

e. Preguntate qué sabe el que está del otro lado del precio. Si no tenés una hipótesis de por qué el mercado está equivocado, no hay edge: hay desconocimiento tuyo.

8 El prompt principal (2 de 2)

PROMPT PRINCIPAL — PARTE 2

ESTIMACIÓN

Recién después de escribir la evidencia, asigná una probabilidad. Nunca antes.

Si tu estimación está a más de 25 puntos del mercado, asumí por defecto que te equivocaste: revisá la condición de resolución y las fechas de tus fuentes antes de seguir.

Redondeá a múltiplos de 5%. Un 53,7% es precisión falsa.

Anotá tu confianza como alta / media / baja. Con confianza baja, no operás: solo registrás.

EDGE

edge_bruto = probabilidad_estimada - precio_YES (en puntos porcentuales)

Estimá la comisión: fee_por_share = shares × feeRate × p × (1-p), con feeRate según categoría (crypto 0,07 / deportes-economía-cultura-clima 0,05 / finanzas-política-tech 0,04 / geopolítica 0).

edge_neto = edge_bruto - comisión estimada - spread

Si edge_neto < 8 puntos porcentuales: descartar y seguir. No negocies el umbral.

TAMAÑO

kelly = (p - c) / (1 - c) · fracción = min(0,25 × kelly ; 0,06)

stake = fracción × bankroll_actual (el de hoy, leído del registro; nunca el inicial).

Si el resultado es menor al mínimo operable del mercado, descartar.

REGLAS QUE NO PODÉS CAMBIAR NI UNA VEZ

- No ejecutás ninguna operación con dinero real sin autorización explícita del usuario para esa operación puntual. En la fase inicial, TODO es simulado.
- No modificás el umbral de 8%, el multiplicador de Kelly, el techo del 6% ni ningún límite de riesgo. Si te parecen mal, lo escribís en el resumen y seguís cumpliéndolos.
- No aumentás el tamaño después de una pérdida para recuperarla. Nunca.
- No abrís una posición correlacionada con otra abierta: mismo evento, mismo partido, misma elección, o mercados que se resuelven con la misma noticia. Antes de proponer, revisá el registro y descartá si hay solapamiento. Dos apuestas al mismo hecho son una apuesta del doble de tamaño.
- No usás información de más de 48 horas para mercados de movimiento rápido. Sin fecha verificable, la fuente no cuenta.
- No operás mercados fuera de los filtros duros aunque el edge parezca enorme.
- Si dos corridas seguidas te dan el mismo mercado con edge creciente, es señal de que tu modelo está sesgado, no de que la oportunidad mejoró. Marcalo y no acumules.
- Si el bankroll simulado cae 25% desde su máximo, paralizás las operaciones nuevas y pedís revisión humana.

REGISTRO Y SALIDA

Al final de cada corrida escribí una fila por decisión (incluidos los descartes y el motivo) en /workspace/polymarket/registro.csv, y un resumen corto: cuántos mercados revisaste, cuántos pasaron cada filtro, las oportunidades con su razonamiento en una línea, y el bankroll simulado. Cerrá siempre aclarando si ejecutaste algo o no.

9 Paper trading: la fase que no te podés saltear

Polymarket no tiene testnet ni sandbox. Pero como los datos son públicos y en tiempo real, la simulación es trivial: el agente hace exactamente lo mismo, con precios reales, y en vez de mandar la orden escribe una fila. Es el mismo sistema, sin el riesgo.

El registro

Un CSV en `/workspace/polymarket/registro.csv`. Estas columnas, sin negociar:

Columna	Qué guarda
`timestamp_utc`	Cuándo se tomó la decisión. En UTC, para poder ordenar.
`mercado`	Slug + pregunta + categoría.
`precio_yes`	El precio al momento de decidir, no el de después.
`prob_estimada`	La probabilidad del agente.
`edge_bruto` / `edge_neto`	En puntos porcentuales, antes y después de comisión y spread.
`fraccion` / `stake_simulado`	Kelly fraccional aplicado y monto simulado.
`bankroll_pre`	Capital simulado antes de la operación.
`resultado`	Vacío hasta que el mercado resuelva. Después: YES / NO.
`pnl`	Ganancia o pérdida simulada, con la comisión descontada.
`bankroll_post`	Capital simulado después. Es el que usa la corrida siguiente.
`razonamiento`	Una línea. Es lo que te deja auditarlo después.
`fuentes`	Links con fecha.

¿CUÁNTO TIEMPO EN SIMULACIÓN?

No te voy a inventar un número mágico, porque no hay uno respaldado. Lo que sí importa: lo que cuenta no son las operaciones *abiertas* sino las **resueltas**. Hasta que el mercado no cierra, no sabés nada.

El criterio práctico: seguí en simulación hasta tener suficientes mercados ya resueltos como para comparar calibración por categoría — de todo lo que el agente dijo "70%", ¿pasó cerca del 70% de las veces? — y hasta haber pasado por al menos un período malo. Un agente que solo vio una racha buena no está probado. Priorizá mercados de resolución corta justamente para que el feedback llegue antes.

TRAMPA CLÁSICA

Si el agente registra el precio después de ver el resultado, o "ajusta" una fila vieja, el registro miente y no te vas a enterar. La fila se escribe cuando se decide, y no se toca más. Solo se completan `resultado` y `pnl` cuando el mercado resuelve.

10 Que corra solo cada 10 minutos

Con la skill guardada, creás una *routine*: le decís qué bot la ejecuta, con qué frecuencia, de dónde saca los datos y qué tiene que devolver. Esta es la estructura del ciclo:

```
cada 10 minutos:
  1. cargar bankroll            ← última fila del registro, nunca un valor fijo
  2. buscar mercados            ← Gamma, closed=false
  3. filtrar                    ← filtros duros; acá muere el 95%
  4. investigar                 ← solo los sobrevivientes
  5. puntuar oportunidades      ← probabilidad + edge neto
  6. aplicar reglas de riesgo   ← 8% / Kelly 1/4 / techo 6% / correlación
  7. simular o ejecutar         ← simular, por defecto
  8. guardar resultados         ← una fila por decisión, incluidos descartes
  9. enviar resumen             ← Telegram
```

ESTO NO LO PUDE CONFIRMAR

La documentación de routines muestra ejemplos diarios ("todos los días de semana a las 8:00") y **no especifica si podés bajar a 10 minutos**, ni si acepta sintaxis cron. Tampoco dice qué pasa si una corrida se solapa con la anterior.

Alternativa si no te deja: el agente tiene terminal y `/workspace` persistente, así que podés dejar un script programado ahí adentro y que la routine sea apenas el resumen cada hora. Ojo: la documentación garantiza que `/workspace` persiste, pero avisa que los paquetes instalados a mano y el estado temporal son reemplazables — o sea, no des por hecho que un cron sobrevive un mantenimiento. Que el script se reinstale solo o chequealo cada tanto.

10 MINUTOS, ¿PARA QUÉ?

Para la mayoría de los mercados, cada 10 minutos es puro gasto: si algo resuelve en 3 semanas, el precio no cambió nada en 10 minutos. La frecuencia alta sirve para mercados que se mueven con noticias. Arrancá cada 30 o 60 minutos, mirá el registro, y bajá la frecuencia solo si ves que se te escapan oportunidades de verdad. Además tené en cuenta que cada corrida consume de tu cuota semanal de uso.

Recordá el límite: un bot corre una sola tarea de computer-use a la vez. Si la corrida de las 10:00 todavía navega cuando arranca la de las 10:10, se van a pelear.

11 Recibir los avisos en Telegram

Grok Bot te notifica por sistema operativo o push al celular, pero **Telegram no viene nativo**. Se arma en dos minutos usando la API oficial de bots de Telegram, que el agente puede llamar desde su terminal:

1. En Telegram, hablale a `@BotFather` y creá un bot. Te da un token.
2. Escribile un mensaje a tu bot nuevo y obtené tu `chat_id` desde `https://api.telegram.org/bot<TOKEN>/getUpdates`.
3. Guardá token y chat_id en un archivo de `/workspace` (no en el prompt).
4. El paso 9 de la rutina hace un POST a `https://api.telegram.org/bot<TOKEN>/sendMessage` con `chat_id` y `text`.

El resumen que querés recibir se parece a esto:

```
3 oportunidades encontradas  ·  847 mercados revisados

Mercado A — "¿X antes del 30/11?"
Precio: 41%   Estimación: 53%   Edge neto: +12 pp
Riesgo propuesto: 2,1% del bankroll ($21)
Razón: dato oficial del 19/09 que el precio todavía no refleja

Mercado B — ...

Bankroll simulado: $1.043  (máximo: $1.120)
No se ejecutó ninguna operación sin aprobación.
```

LO QUE HACE ÚTIL EL RESUMEN

Que incluya los **descartes** y el bankroll. "847 revisados, 3 pasaron" te dice que los filtros funcionan. Un resumen que solo trae ganadores es un resumen que te está vendiendo algo.

SEGURIDAD

El token de Telegram queda en la computadora en la nube, que comparten todos tus bots. Cualquiera de ellos lo puede usar. No pongas ahí nada más sensible que eso — y mucho menos una clave privada de wallet.

12 Prompts para auditar y mejorar el agente

Corrélos vos, a mano, cada tanto. No los pongas dentro de la rutina automática: un agente que se audita y se corrige solo, sin nadie mirando, deriva.

AUDITORÍA DE DECISIONES

```
Leé /workspace/polymarket/registro.csv y auditá tus últimas 30 decisiones. Para cada una, compará tu probabilidad estimada contra el resultado real, solo en los mercados ya resueltos. Después contestame:

1. CALIBRACIÓN: agrupá por rango (50-60%, 60-70%, 70-80%, 80%+). En cada rango, ¿qué proporción se cumplió realmente? Si en el grupo del 70% acertaste el 45%, estás sobreconfiado y quiero el número, no una explicación.

2. PÉRDIDAS: listá las 5 peores. Para cada una, ¿el error fue la estimación, la lectura de la condición de resolución, la fuente, o el tamaño? Una sola categoría por pérdida.

3. CATEGORÍAS: ¿en qué tipo de mercado te va sistemáticamente peor? Decime cuántas operaciones resueltas tenés en cada categoría junto al resultado. Con menos de 10, decí "muestra insuficiente" en vez de una conclusión.

4. SESGOS: ¿hay un patrón? Por ejemplo: siempre estimás más alto que el mercado, o te entusiasmás con noticias recientes, o tus peores operaciones vienen de una misma fuente.

No propongas cambios todavía. Solo mostrame los números.
```

REVISIÓN DE ESTRATEGIA (DESPUÉS DE LA AUDITORÍA)

```
En base a la auditoría, proponé como máximo UN cambio a la estrategia. Reglas para proponerlo:

· Tiene que estar respaldado por al menos 20 operaciones resueltas. Con menos, la respuesta correcta es "no hay evidencia suficiente para cambiar nada".

· No podés proponer bajar el umbral del 8%, subir el techo del 6% ni aumentar el multiplicador de Kelly. Esos van en una sola dirección: más conservadores.

· Decime explícitamente qué tendría que pasar para que el cambio resulte un error.

· Si el cambio es "operar más", asumí que está mal y explicame por qué no.

El cambio no se aplica hasta que yo lo apruebe.
```

EL SESGO MÁS CARO

Cambiar la estrategia con pocos resultados. Con 8 operaciones no hay señal: hay ruido. Un agente al que le pedís que mejore algo siempre va a encontrar algo que "mejorar", y casi siempre en la dirección de operar más. De ahí que el prompt lo obligue a decir "muestra insuficiente", y que el único cambio posible sea hacia el lado conservador.

13 Qué NO hacer

No asumas que $50 → $5.000 es reproducible. Es una afirmación sin verificar, de 48 horas. No es un resultado esperable.
No le des acceso a una wallet con capital importante. Lo que dejás ahí es lo que podés perder. Usá session keys, que no pueden retirar fondos.
No lo dejes cambiar sus límites de riesgo. Si el agente puede reescribir el 6%, el 6% no existe.
No te creas una estimación sin mirar las fuentes. Un razonamiento bien escrito no es un razonamiento correcto.
No operes mercados sin liquidez. El edge que no podés ejecutar ni cerrar no es edge.
No automatices dinero real desde el primer día. Simulación primero, con operaciones ya resueltas, y recién después plata — con aprobación por operación.

14 Qué te queda armado

Pieza	Dónde vive
Agente con navegador, terminal y almacenamiento	Grok Bot + su computadora en la nube
Rutina que corre sola con tu compu apagada	Routine programada
Prompt con la lógica de análisis y riesgo	Skill guardada (páginas 9 y 10)
Registro de cada decisión, incluidos los descartes	`/workspace/polymarket/registro.csv`
Paper trading con precios reales	Misma rutina, sin ejecutar
Alertas con resumen y bankroll	Telegram
Controles de riesgo que el agente no puede tocar	Umbral 8% · Kelly 1/4 · techo 6% · anti-correlación · freno por drawdown

EL ÚNICO PASO QUE IMPORTA AHORA

Creá el agente, pegale el prompt de las páginas 9 y 10, y dejalo correr en simulación. Nada más. Si dentro de unas semanas el registro muestra que está bien calibrado, recién ahí tiene sentido hablar de plata real. Y si muestra que no, te enteraste gratis — que es exactamente para lo que sirve esta fase.

Esta guía es material educativo sobre cómo se construye un agente, no asesoramiento financiero. Los mercados de predicción pueden hacerte perder todo el capital que pongas, y no están disponibles legalmente en todos los países.



NodeStudio

Automatizamos procesos de negocio con IA y sistemas a medida.

nodestudioai.com