Facsímil 08

La ciencia de los datos

Capítulo 07

Análisis aplicado: experimentos, causalidad y decisión

686f6c61 · 2026

Capítulo 07: Análisis aplicado: experimentos, causalidad y decisión

Entrando en el tema

En el capítulo anterior aprendimos a operar datos: ventanas, drift, trazas, SLOs, gates y postmortems. Eso nos protege de tomar decisiones con datos rotos. Ahora damos el siguiente paso: **decidir si una acción cambia algo**.

Esta es una de las fronteras más importantes para cualquier persona que trabaja con IA. Un modelo puede predecir que un estudiante no resolverá su trámite a tiempo. Pero otra pregunta mucho más difícil es: ¿qué acción aumenta realmente la probabilidad de resolverlo? ¿Enviar una plantilla? ¿Escalarlo? ¿Cambiar el orden de cola? ¿Dar un mensaje más claro? ¿Nada?

Al terminar deberías poder hacer esto:

RESULTADO DE APRENDIZAJE	EVIDENCIA DE QUE LO SABES HACER
Separar predicción de intervención.	No usas un score predictivo como prueba de efecto causal.
Escribir una pregunta causal.	Distingues tratamiento, control, resultado, unidad y población.
Diseñar un experimento mínimo.	Defines asignación, métrica primaria, guardrails, tamaño y criterio de decisión.
Leer un A/B test como artefacto de ingeniería.	Buscas SRM, balance, slices, intervalos y trazabilidad.
Diseñar una plataforma mínima de experimento.	Separas flag, contexto, exposición, métricas, análisis y release gate.
Calcular readiness estadístico.	Escribes MDE, alpha, potencia, muestra y política de peeking.
Entender ATE, CATE y uplift.	Sabes cuándo importa el efecto medio y cuándo importa el efecto por segmento.
Auditar datos observacionales con cuidado.	No cierras una conclusión causal si falta solapamiento o hay confounding.
Generar una decisión reproducible.	Dejas reportes, scorecards y una decisión versionada.

Esta lista no pretende convertirte en estadístico en diez minutos. Pretende darte una brújula de ingeniería: antes de tocar un modelo, un prompt, un RAG o una política automática, debes saber qué decisión vas a permitirte tomar con los datos. Hay experimentos que solo sirven para aprender, otros que sirven para bloquear una release y otros que solo justifican una conversación con más muestra. Confundir esos tres niveles es una fuente silenciosa de malas decisiones.

En este capítulo, cada fórmula y cada artefacto práctico intentan responder a una pregunta operativa. Si una estimación no cambia qué publicas, qué bloqueas, qué repites o qué escalas a revisión humana, todavía no está haciendo trabajo de ingeniería. Por eso aparecen contratos, catálogos de métricas, eventos de exposición y decisiones versionadas junto a causalidad: el número importa, pero el circuito que lo produce importa igual o más.

La frase central:

> Predecir responde qué puede pasar. Causalidad responde qué cambiaría si actuamos.

La escena: el descuento que parecía funcionar

Imagina que tenemos un asistente para soporte académico. El sistema predice qué casos tienen más probabilidad de resolverse tarde. Un equipo propone enviar una plantilla guiada a ciertos casos antes de que el operador responda. Al mirar datos históricos, los casos que recibieron plantilla parecen resolverse mucho más.

La tentación es decir: “perfecto, la plantilla funciona”. Pero quizá la plantilla se mandaba sobre todo a casos de matrícula, que ya tenían más probabilidad de resolverse. O quizá se mandaba a estudiantes con expedientes más completos. O quizá los operadores más expertos usaban más la plantilla y también resolvían mejor sin ella.

El análisis aplicado empieza cuando frenamos esa conclusión rápida y escribimos la pregunta correcta:

PREGUNTA FLOJA	PREGUNTA ÚTIL
¿Quién resuelve mejor?	¿Qué cambia si asignamos la plantilla?
¿Qué variable predice resolución?	¿Qué acción modifica la resolución?
¿La gente con plantilla resolvió más?	¿Casos comparables resolverían más con plantilla que sin plantilla?
¿El modelo lo explica?	¿El diseño permite estimar el efecto?

La diferencia entre esas dos columnas parece lingüística, pero cambia todo el proyecto. En la columna izquierda todavía estamos describiendo el mundo: vemos correlaciones, rankings, grupos que se comportan mejor y variables que anticipan un resultado. En la columna derecha empezamos a diseñar una intervención: definimos qué se cambia, sobre quién, cuándo madura el resultado y qué comparación sería justa.

Esta es una de las razones por las que la causalidad encaja tan bien en ingeniería de IA. Un sistema predictivo puede ordenar una cola y ser útil. Pero en cuanto el sistema actúa, prioriza, recomienda, escala, resume, bloquea o decide qué herramienta usar, ya no basta con predecir. En ese momento necesitas saber si la acción mejora algo o si simplemente estás actuando sobre los casos que ya eran más fáciles.

Qué no es análisis causal

Análisis causal no es mirar una correlación y escribir una historia convincente. Las historias pueden ayudar a formular hipótesis, pero no convierten una asociación en efecto.

Tampoco es una capa que arregle datos malos por sí sola. Si la asignación está rota, si el resultado se mide tarde, si cambió el producto a mitad del experimento o si falta trazabilidad, la fórmula no salva la decisión. Por eso este capítulo depende tanto del capítulo 06 sobre DataOps.

Y no es interpretabilidad. SHAP, importancia de variables, PDP, ICE o trazas pueden ayudarnos a preguntar mejor, pero no prueban por sí solas que una acción cause un resultado. Si una variable aparece como importante, quizá sea causa, quizá sea proxy, quizá sea síntoma o quizá esté recogiendo una regla de negocio.

Qué sí es análisis aplicado para IA

Análisis aplicado es convertir datos en una decisión defendible. En IA suele aparecer en tres momentos:

MOMENTO	PREGUNTA	EJEMPLO
Antes de publicar	¿Este cambio mejora algo real?	Nuevo prompt, nuevo ranking, nueva política de revisión.
Después de publicar	¿La mejora se mantiene?	Ventana post-release comparada con control o holdout.
Al priorizar acciones	¿Dónde conviene intervenir?	Casos donde una plantilla cambia la resolución, no donde ya iba a resolverse.

En equipos reales, estas tres situaciones se mezclan. Un nuevo prompt puede mejorar la tasa de respuesta aceptada, pero quizá empeore citas, latencia o coste. Un nuevo reranker puede recuperar mejores documentos, pero solo para consultas largas. Una regla de revisión humana puede reducir errores graves, pero saturar a un equipo operativo. La causalidad aplicada no busca una medalla para la variante ganadora; busca separar mejora real, coste operativo y riesgo residual.

Rubin formalizó el enfoque de resultados potenciales: para cada unidad imaginamos el resultado bajo tratamiento y bajo control, aunque solo observemos uno.[1] Holland popularizó la idea de que la inferencia causal se enfrenta a un problema fundamental: no podemos observar simultáneamente ambos resultados potenciales para la misma unidad.[2] Pearl desarrolló un marco gráfico y el operador `do` para razonar sobre intervención, ajuste y supuestos causales.[3]

Fecha de corte: **7 de junio de 2026**. Fuentes consultadas ese día: literatura clásica de causalidad, Microsoft ExP, CUPED, DoWhy, EconML, OpenFeature, LaunchDarkly, Statsig, GrowthBook y trabajos sobre calidad de experimentos a escala. Lo estable es la diferencia entre asociación, intervención, contrafactual y decisión. Lo que cambia son herramientas, plataformas y automatizaciones.

Predicción e intervención no responden lo mismo

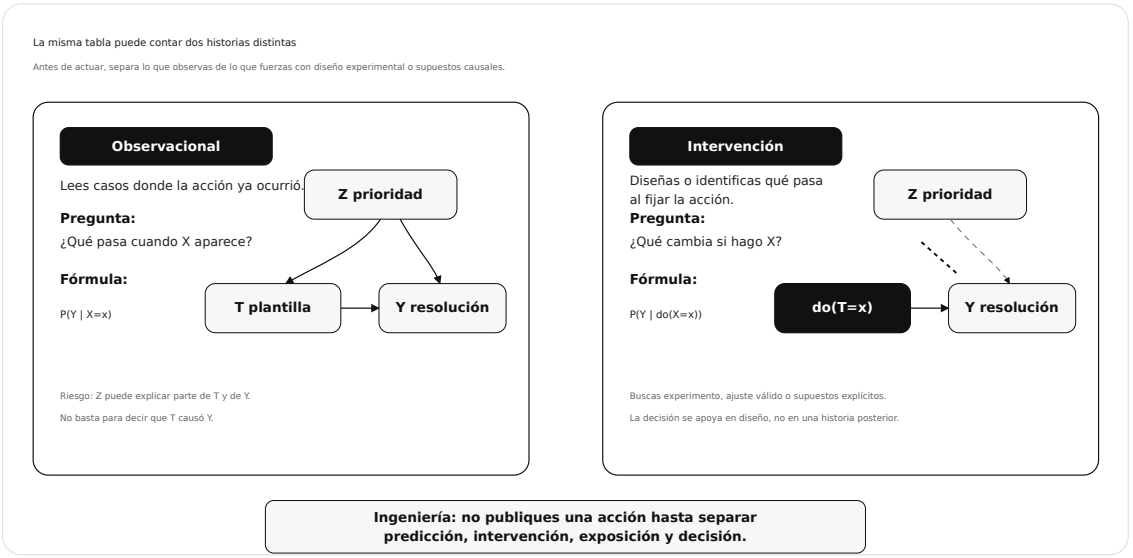
La confusión más habitual cabe en dos expresiones parecidas:

$$P(Y \mid X = x)$$

$$P(Y \mid do(X = x))$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
Y	Resultado que medimos.	Caso resuelto.
X	Variable o acción que miramos.	Recibir plantilla.
x	Valor concreto de X.	`plantilla = sí`.
$P(Y \mid X = x)$	Probabilidad observada dado que X ocurrió.	Resolución entre casos que recibieron plantilla.
$do(X = x)$	Intervención que fija X desde el diseño.	Asignar plantilla por experimento.
$P(Y \mid do(X = x))$	Probabilidad bajo intervención.	Resolución si forzamos plantilla en casos comparables.

La primera expresión es predictiva u observacional. Nos dice qué pasa entre casos donde X aparece. La segunda es causal. Nos pregunta qué pasa si intervenimos.



> El operador `do` no es un adorno matemático: marca que la acción se fija por diseño o por un supuesto causal defendible.

Para un ingeniero de IA, la diferencia es práctica:

SI HACES...	NECESITAS...	POR QUÉ
Ordenar casos por riesgo	Predicción bien evaluada.	Quieres anticipar qué ocurrirá.
Cambiar una política	Efecto causal o experimento.	Quieres saber qué cambia por actuar.
Elegir a quién enviar una acción	Uplift o CATE.	No basta saber quién tiene más probabilidad de éxito.
Publicar un cambio de producto	Diseño experimental y guardrails.	Debes proteger métricas secundarias y operación.

El problema de los resultados potenciales

La notación de resultados potenciales escribe dos mundos para cada unidad:

$$Y_i(1)$$

$$Y_i(0)$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
i	Unidad analizada.	Ticket `u014`.
$Y_i(1)$	Resultado de la unidad si recibe tratamiento.	Resolvería con plantilla.
$Y_i(0)$	Resultado de la unidad si recibe control.	Resolvería sin plantilla.
1	Tratamiento activo.	`treatment`.
0	Control.	`control`.

El problema es que para una misma unidad solo observamos una de las dos columnas. Si `u014` recibió plantilla, vemos $Y_{014}(1)$, pero no vemos $Y_{014}(0)$. Ese resultado no observado es el contrafactual.

Por eso no medimos el efecto individual real de cada caso. Estimamos efectos agregados bajo supuestos:

$$ATE = E[Y(1) - Y(0)]$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
ATE	Efecto medio del tratamiento.	+0.25 en tasa de resolución.
$E[\cdot]$	Esperanza o promedio poblacional.	Media sobre los casos del experimento.
$Y(1)$	Resultado bajo tratamiento.	Resolver con plantilla.
$Y(0)$	Resultado bajo control.	Resolver sin plantilla.

Si aleatorizamos bien, el grupo control actúa como aproximación del mundo sin tratamiento y el grupo tratamiento como aproximación del mundo con tratamiento.

Un A/B test como sistema de ingeniería

Kohavi, Longbotham, Sommerfield y Henne trataron los experimentos online como práctica central para tomar decisiones en la web, pero también documentaron problemas habituales de diseño, métrica y ejecución.[4] Microsoft ExP describe una plataforma de experimentación

con portal, servicio de ejecución, procesamiento de logs y análisis como piezas separadas de una arquitectura escalable.[5]

Esto es importante: un A/B test no es una tabla con `control` y `treatment`. Es un sistema.

PIEZA	PREGUNTA DE INGENIERÍA	FALLO TÍPICO
Unidad	¿Qué se asigna?	Mezclar usuario, sesión y ticket.
Asignación	¿Cómo se decide control o tratamiento?	Reparto roto o no reproducible.
Persistencia	¿La unidad conserva variante?	Un usuario cambia de grupo entre eventos.
Métrica primaria	¿Qué decide el experimento?	Optimizar una proxy cómoda.
Guardrails	¿Qué no puede empeorar?	Mejorar resolución subiendo coste o latencia.
Logging	¿Qué evento demuestra exposición?	Medir casos que nunca vieron la intervención.
Análisis	¿Qué estimador y regla de parada usamos?	Mirar cada día y parar cuando conviene.
Decisión	¿Qué hacemos con el resultado?	Publicar sin contrato ni evidencia.

La tabla se puede leer como una arquitectura mínima. Si falla la unidad, el resto del análisis queda contaminado porque no sabemos qué estamos comparando. Si falla la asignación, no sabemos si la variante llegó a quien debía. Si falla la exposición, contamos unidades que nunca vivieron el tratamiento. Si falla el logging, no podemos reconstruir la historia. Y si falla la decisión, el experimento se convierte en un informe bonito que nadie sabe usar.

Esta mirada es especialmente importante en IA porque el tratamiento suele ser más difuso que en una web clásica. No siempre cambias un botón. A veces cambias el prompt, el modelo, el proveedor, el `temperature`, el chunking, la política de tools, el umbral de revisión o la plantilla de salida. Si no registras esa versión exacta, mañana no podrás repetir el experimento ni explicar por qué una variante funcionó.

Plan de análisis antes de tocar resultados

Un experimento serio se empieza escribiendo antes qué se va a mirar. Esto no es solemnidad académica: es protección contra cambiar la pregunta cuando ya hemos visto la respuesta.

El cuaderno del facsímil incluye `analysis_plan.json`. Ese archivo fija hipótesis, población, tratamiento, control, métrica primaria, guardrails, slices a reportar, exclusiones y reglas de decisión. También incluye `metric_catalog.json`, porque una métrica sin definición estable se convierte en una palabra elástica.

PIEZA DEL PLAN	PREGUNTA QUE RESPONDE	EJEMPLO
Hipótesis	¿Qué creemos que cambiará?	La plantilla guiada aumenta resolución a 7 días.
Población	¿Sobre quién vale la decisión?	Casos académicos con trazabilidad completa.
Unidad	¿Qué se asigna y analiza?	`unit_id`.
Tratamiento	¿Qué cambia exactamente?	Plantilla guiada + contexto documental revisado.
Control	¿Contra qué comparamos?	Flujo actual.
Métrica primaria	¿Qué decide el experimento?	`resolved`.
Ventana	¿Cuándo madura la métrica?	`day_7`.
Guardrails	¿Qué no puede empeorar?	Feedback, latencia, coste, cita válida.
Exclusiones	¿Qué se elimina antes del análisis?	Sin exposure event, fallback no etiquetado, unidad inestable.
Regla de decisión	¿Qué significa `pass`, `review` o `block`?	Publicar solo con readiness y guardrails en `pass`.

Un buen plan de análisis es una defensa contra uno mismo. Cuando ya has visto una gráfica favorable, todo el mundo encuentra razones para cambiar la ventana, mover una exclusión, mirar otro segmento o explicar que la métrica importante era otra. El plan escrito antes reduce esa elasticidad. No elimina el juicio, pero deja claro cuándo estás ejecutando el análisis pactado y cuándo estás explorando.

En una organización, además, el plan de análisis permite repartir responsabilidades. Producto puede defender la hipótesis, ingeniería puede defender la instrumentación, datos puede defender la métrica y operación puede defender los guardrails. Sin esa separación, el experimento acaba siendo una reunión donde cada persona trae una intuición distinta y ninguna se puede auditar.

El detalle importante es “antes”. Si primero miramos el resultado y luego elegimos la métrica que más favorece a la variante, no estamos evaluando; estamos buscando una justificación.

`validate_experiment_design.py` revisa que el plan, catálogo y contrato de flag encajen. No calcula efecto. Hace algo anterior y muy de ingeniería: comprueba si el experimento está suficientemente definido para poder medir.

El catálogo de métricas cumple otra función: evita que dos personas usen la misma palabra para cosas distintas. En un proyecto real, `resolved` no puede significar "el operador cerró el ticket" un día y "el estudiante no volvió a escribir" al siguiente. Si la métrica decide publicación, tiene que tener unidad, ventana, dirección y definición operativa.

CAMPO EN `METRIC_CATALOG.JSON`	QUÉ OBLIGA A DECIDIR	EJEMPLO DEL CUADERNO
`name`	Nombre estable para código, SQL y reporte.	`resolved`.
`type`	Papel de la métrica.	`primary`, `guardrail`, `diagnostic`.
`unit`	Grano de medición.	`unit_id`.
`window`	Cuándo se considera madura.	`day_7` o `exposure`.
`direction`	Qué significa mejorar.	`higher_is_better` o `lower_is_better`.
`definition`	Interpretación que debe sobrevivir al cambio de equipo.	Caso resuelto correctamente dentro de la ventana observada.

Este catálogo parece pequeño, pero evita una cantidad enorme de ambigüedad. En sistemas de IA, una misma palabra puede esconder métricas distintas: "calidad" puede ser aceptación del usuario, evaluación humana, similitud con una respuesta de referencia, precisión de citas, ausencia de toxicidad o reducción de escalados. Si no se escribe la semántica, cada dashboard termina midiendo una cosa distinta con el mismo nombre.

Esto parece burocrático hasta que falla. Si un experimento mejora `answer_accepted` pero empeora `citation_valid`, o si `latency_ms` se mide desde puntos distintos del backend, no tienes una decisión: tienes una conversación interminable. El catálogo corta esa ambigüedad antes de ejecutar.

Plataforma experimental para sistemas de IA

Cuando un equipo de IA madura, el experimento deja de vivir en una hoja de cálculo. Se convierte en una plataforma con piezas separadas. OpenFeature define una especificación abierta para feature flags con una API común y proveedores intercambiables.[6] La

especificación de `evaluation context` describe el contexto que viaja con una evaluación de flag, incluyendo un `targeting key` que identifica la unidad sobre la que se evalúa la variante.[7]

La idea que nos interesa no es la marca de la herramienta. Es el contrato de plataforma:

COMPONENTE	QUÉ HACE	QUÉ DEBE QUEDAR TRAZADO
Feature flag	Decide qué variante recibe una unidad.	`flag_key`, versión, proveedor y regla.
Evaluation context	Aporta atributos para evaluar la flag.	`targeting_key`, segmento, región, canal, entorno.
Assignment service	Hace persistente la variante.	Unidad, variante, timestamp, hash de regla.
Exposure event	Registra que la unidad vio la variante.	Evento de exposición, no solo asignación teórica.
Metrics layer	Define métricas con una semántica estable.	Nombre, ventana, unidad, filtros, owner.
Warehouse	Une exposición, eventos y métricas.	Tablas, particiones, linaje y retrasos de datos.
Analysis service	Calcula efecto, intervalos, slices y guardrails.	Método, parámetros, fecha, versión de contrato.
Release gate	Convierte análisis en acción.	`pass`, `review`, `block`, rollout o rollback.

La separación por capas tiene una ventaja muy concreta: permite depurar. Si el resultado parece extraño, no necesitas discutir todo a la vez. Puedes mirar si la flag asignó bien, si el contexto llevaba el `targeting_key`, si la exposición se emitió, si el warehouse unió tarde los eventos, si la métrica se calculó con la ventana correcta o si el release gate interpretó mal el estado. Sin capas, cada incidente se convierte en una niebla.

En proyectos de IA esto también ayuda a comparar proveedores y modelos sin reescribir todo el sistema. El proveedor de LLM puede cambiar, el framework de agentes puede cambiar y el motor de RAG puede cambiar, pero el contrato experimental debería seguir pidiendo lo mismo: unidad, variante, exposición, métrica, evidencia y decisión.

LaunchDarkly describe experimentación conectando métricas a flags y configuraciones para medir cambios en comportamiento, y menciona A/A testing para validar instrumentación antes de un A/B real.[8] GrowthBook se presenta como plataforma open source de feature flags y experimentación, con énfasis en métricas sobre datos existentes y transparencia de consultas.[9] Statsig documenta opciones avanzadas como duración de asignación, periodo de análisis, covariables, sequential testing y correcciones por múltiples comparaciones.[10]

En la práctica aparecen más opciones: Eppo encaja bien cuando el equipo quiere experimentación apoyada en su propio stack de datos y feature flags de bajo acoplamiento.[11] PostHog combina flags, analítica de producto y experimentos dentro de una misma plataforma, útil cuando un equipo pequeño quiere cerrar el circuito sin montar una plataforma propia desde cero.[12] Optimizely Feature Experimentation se sitúa más cerca de una plataforma enterprise con SDKs, APIs y experimentación en múltiples superficies.[13] Evidently no es una plataforma de causalidad, pero ayuda a vigilar drift, calidad de datos y métricas de sistemas de IA antes o después de un experimento.[14]

En IA, esta arquitectura tiene matices propios:

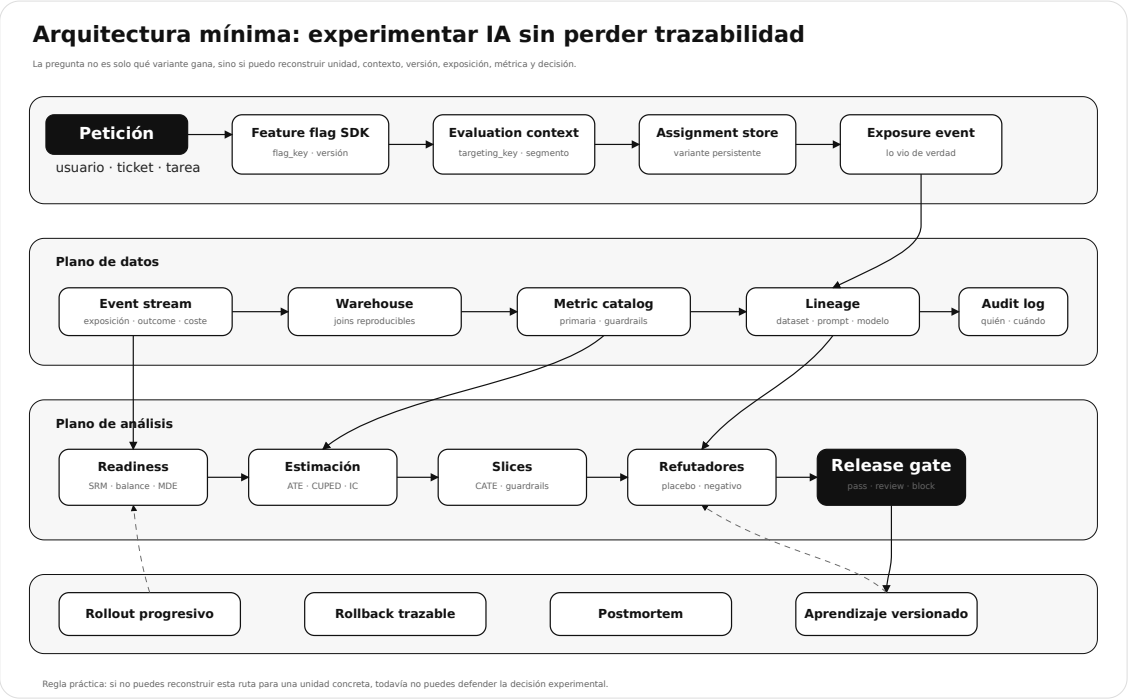
CASO DE IA	QUÉ SE EXPERIMENTA	RIESGO DE INGENIERÍA
Nuevo prompt	Variante textual o plantilla.	La exposición debe registrar versión exacta del prompt.
Nuevo modelo	Modelo, proveedor o configuración.	Latencia, coste y fallback importan tanto como calidad.
RAG	Retriever, reranker, chunking o fuente.	Cambia el contexto recuperado, no solo la respuesta.
Agente	Política de herramientas o aprobación.	La unidad puede ser tarea, conversación o usuario.
Recomendación	Ranking o regla de diversidad.	Un usuario puede afectar inventario, cola o experiencia de otros.
Moderación de cola	Umbral o política de revisión.	Guardrails humanos y operativos son obligatorios.

Fíjate en que “tratamiento” no significa siempre “modelo A contra modelo B”. En un RAG puede ser la forma de partir documentos; en un agente puede ser exigir aprobación humana antes de una herramienta; en un sistema local puede ser una cuantización; en un clasificador puede ser el umbral de derivación. Esta amplitud obliga a documentar la intervención con precisión. Si el tratamiento es ambiguo, el efecto también lo será.

La pregunta de ingeniería no es “¿qué librería uso?”. Es esta:

```
¿Puedo reconstruir qué unidad recibió qué variante,
con qué contexto, bajo qué versión, y qué métricas maduraron después?
```

Si la respuesta es no, todavía no hay experimento publicable.



La plataforma separa ejecución, datos, análisis y decisión. Esa separación evita confundir “la variante se asignó” con “la variante se expuso, maduró y permite decidir”.

Unidad, exposición e interferencia

La unidad de asignación es una decisión de diseño. Si asignas por sesión, pero el usuario vuelve varias veces, puede ver control y tratamiento. Si asignas por ticket, pero el operador aprende de una variante y cambia su comportamiento en otros tickets, hay contaminación entre unidades. Si asignas por empresa, tendrás menos unidades pero menos mezcla entre grupos.

UNIDAD	SIRVE CUANDO	CUIDADO
Usuario	La experiencia se mantiene en el tiempo.	Varias sesiones deben conservar variante.
Sesión	La intervención dura solo una visita.	No sirve para métricas que maduran por usuario.
Ticket	Cada caso es independiente.	Un mismo operador puede influir varios tickets.
Empresa	Hay efecto compartido entre usuarios.	Menos muestra y más varianza.
Conversación	Agentes o asistentes conversacionales.	Una conversación puede contener varias tareas.

La unidad no se elige por comodidad, sino por independencia y por consecuencia. Si el efecto de una variante se queda dentro de una sesión, la sesión puede servir. Si el efecto acompaña a un usuario durante días, una sesión es demasiado pequeña. Si varios usuarios de una misma empresa comparten políticas, documentos o límites, quizá la empresa sea la unidad honesta. La muestra más grande no siempre es la más creíble.

En IA conversacional aparece una trampa adicional: una conversación puede contener varias tareas y cada tarea puede activar herramientas distintas. Si asignas por mensaje, puedes mezclar variantes dentro de la misma conversación. Si asignas por usuario, quizá arrastres aprendizaje o memoria. Si asignas por tarea, necesitas detectar bien dónde empieza y termina. Esta decisión parece técnica, pero cambia qué puedes afirmar al final.

La exposición también importa. Asignar una variante no significa que la unidad la haya recibido. En un sistema de IA puede haber fallback, timeout, caché, error de proveedor, ruta alternativa o una regla de permisos que impide mostrar la variante. Por eso el contrato del cuaderno exige `experiment_exposure`: una decisión experimental se analiza sobre unidades expuestas, no solo asignadas.

Interferencia significa que una unidad puede afectar el resultado de otra. En producto digital aparece en marketplaces, colas, rankings, soporte compartido y sistemas con capacidad limitada. En IA aparece también cuando un agente consume herramientas compartidas, cambia una cola humana, modifica documentos o prioriza casos que compiten por el mismo recurso. Si hay interferencia fuerte, el A/B clásico por usuario puede quedarse corto y quizá necesites asignar por clúster, equipo, cola o ventana temporal.

El cuaderno del facsímil incluye `cluster_interference_events.csv`. La idea es sencilla: si dentro de un mismo equipo conviven control y tratamiento, y el operador aprende de la variante nueva, el control puede contaminarse. En ese caso quizá la unidad correcta no sea `unit_id`, sino `cluster_id`.

SEÑAL	LECTURA
Cluster con control y treatment mezclados	Posible contaminación de comportamiento.
Mismo operador en ambas variantes	Puede aprender de treatment y aplicarlo a control.
Cola compartida	Una variante puede cambiar tiempos de la otra.
Recurso limitado	La mejora de un grupo puede desplazar capacidad.

La interferencia es incómoda porque rompe una suposición que solemos dar por hecha: que cada unidad vive su tratamiento aislada del resto. En una cola de soporte, eso casi nunca es del todo cierto. Si una variante libera tiempo de operadores, puede mejorar también al

control. Si una variante consume más revisión humana, puede empeorar al control. Si un agente aprende una estrategia que un operador imita, la frontera entre grupos se vuelve borrosa.

Una regla práctica:

```
Si las unidades comparten operador, inventario, cola, aula, empresa o herramienta limitada,
pregunta si deberías asignar por clúster.
```

Si asignas por clúster, la pregunta cambia ligeramente. Ya no preguntas solo por tickets individuales; preguntas por equipos, empresas, aulas o colas completas. Donner y Klar desarrollan el diseño y análisis de ensayos aleatorizados por clúster, donde la unidad de asignación y la dependencia dentro de grupo son parte central del análisis.[15] Una forma pedagógica de verlo es calcular primero el resultado medio de cada clúster y después comparar clústeres:

$$\widehat{ATE}_{cluster} = \frac{1}{K_T}\sum_{k \in T} \bar{Y}_k - \frac{1}{K_C}\sum_{k \in C} \bar{Y}_k$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
k	Clúster completo.	`team_a`, `team_b`, `team_c`.
$\bar{Y}_k$	Resultado medio dentro del clúster k.	Resolución media del equipo.
K_T	Número de clústeres en tratamiento.	Equipos asignados a la variante nueva.
K_C	Número de clústeres en control.	Equipos asignados al flujo actual.

La ventaja es que reduces contaminación entre unidades. El precio es que tienes menos observaciones efectivas: cien tickets dentro de tres equipos no equivalen a cien unidades independientes si el equipo comparte operador, cola y aprendizaje. Por eso la decisión de unidad no es un detalle técnico pequeño; cambia la potencia, el coste y la credibilidad del experimento.

Una forma clásica de explicarlo viene del muestreo por clúster. Kish popularizó el *design effect* o efecto de diseño para aproximar cuánto se infla la varianza cuando las observaciones dentro de un clúster se parecen entre sí.[16] Una versión habitual se escribe:

$$DEFF = 1 + (m - 1)\rho$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$DEFF$	Efecto de diseño.	`1.45` .
m	Tamaño medio del clúster.	10 tickets por equipo.
ρ	Correlación intraclúster aproximada.	`0.05` .

En palabras: si los casos dentro de un mismo equipo se parecen, cada ticket aporta menos información independiente de lo que parece. Con $m = 10$ y $\rho = 0.05$, el efecto de diseño es $1 + 9 \cdot 0.05 = 1.45$. Una lectura rápida del tamaño efectivo sería:

$$n_{eff} \approx \frac{n}{DEFF}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
n_{eff}	Tamaño muestral efectivo aproximado.	100 tickets se parecen a unos 69 casos independientes.
n	Número de observaciones registradas.	100 tickets.
$DEFF$	Factor que descuenta dependencia por clúster.	`1.45` .

En palabras: no estás tirando datos; estás dejando de contarlos dos veces. Para un experimento de IA con operadores, aulas, empresas, colas o herramientas compartidas, esta idea evita una trampa muy habitual: diseñar el A/B por evento cuando el comportamiento real se mueve por grupo.

El estimador básico en un A/B test es diferencia de medias:

$$ATE = \bar{Y}_T - \bar{Y}_C$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$\widehat{ATE}$	Estimación del efecto medio.	`0.25`.
$\bar{Y}_T$	Media del resultado en tratamiento.	`0.833333` resueltos.
$\bar{Y}_C$	Media del resultado en control.	`0.583333` resueltos.
T	Grupo tratamiento.	Casos con plantilla guiada.
C	Grupo control.	Casos sin plantilla nueva.

En el cuaderno, el efecto observado es:

$$ATE = 0.833333 - 0.583333 = 0.25$$

Esto suena fuerte, pero no basta. Hay que mirar incertidumbre. La forma siguiente es la aproximación habitual del error estándar de una diferencia de medias con grupos independientes, base de los intervalos y tests introductorios que verías en un curso clásico de inferencia estadística.[17]

$$\widehat{SE(ATE)} = \sqrt{\frac{s_T^2}{n_T} + \frac{s_C^2}{n_C}}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$\widehat{SE}$	Error estándar de la estimación.	`0.186339`.
s_T^2	Varianza del resultado en tratamiento.	Varianza de `resolved` en treatment.
s_C^2	Varianza del resultado en control.	Varianza de `resolved` en control.
n_T	Número de unidades en tratamiento.	`12`.
n_C	Número de unidades en control.	`12`.

Y un intervalo aproximado:

$$IC_{95\%} = \widehat{ATE} \pm 1.96 \cdot \widehat{SE(ATE)}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$IC_{95\%}$	Intervalo de confianza aproximado al 95%.	`[-0.115224, 0.615224]`.
1.96	Multiplicador normal aproximado para 95%.	Valor estándar.
ATE	Efecto estimado.	`0.25`.
$\widehat{SE}$	Error estándar.	`0.186339`.

La lectura honesta es: señal positiva, todavía imprecisa. Publicar automáticamente sería precipitado. Repetir o ampliar muestra es más defendible.

Calidad del experimento antes de mirar el resultado

Antes de celebrar un efecto, un ingeniero revisa si el experimento merece confianza. En el cuaderno lo hacemos con tres controles: SRM, balance y guardrails.

SRM significa *sample ratio mismatch*: el reparto observado no coincide con el reparto esperado. Si diseñamos 50/50 y aparece 70/30, quizá la asignación, tracking o filtrado está fallando. Nie y colaboradores describen validaciones automáticas de aleatorización y detección de SRM en plataformas de experimentación a escala.[18]

Una comprobación clásica usa chi-cuadrado:

$$\chi^2 = \sum_{g \in G} \frac{(O_g - E_g)^2}{E_g}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
χ^2	Estadístico de desajuste.	`0.0`.
G	Grupos del experimento.	`control`, `treatment`.
O_g	Conteo observado en grupo g.	`12`.
E_g	Conteo esperado en grupo g.	`12`.

El balance revisa si las covariables previas al tratamiento son parecidas. Si los grupos ya eran distintos antes de actuar, el efecto puede mezclarse con composición.

Una medida simple es la diferencia de medias estandarizada. En estudios observacionales y emparejamiento con propensity score se usa mucho como diagnóstico de balance, porque expresa diferencias de covariables en unidades comparables.[19]

$$SMD = \frac{\bar{X}_T - \bar{X}_C}{s_{pooled}}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
SMD	Diferencia estandarizada entre grupos.	`0.0`.
$\bar{X}_T$	Media de covariable previa en tratamiento.	Tareas previas en treatment.
$\bar{X}_C$	Media de covariable previa en control.	Tareas previas en control.
s_{pooled}	Desviación típica combinada.	Variación compartida entre grupos.

Los guardrails son métricas que no deciden la victoria, pero sí pueden impedir publicar. En el cuaderno, la intervención mejora resolución, pero vigilamos coste, latencia y feedback negativo.

Esta parte es donde muchos experimentos “ganadores” dejan de serlo. Una variante que resuelve más tickets puede ser peor si duplica coste, degrada latencia, aumenta errores de cita o desplaza trabajo a revisión humana. En IA, los guardrails no son decoración ética ni prudencia abstracta: son métricas de supervivencia del sistema. Si el usuario recibe mejores respuestas pero el equipo no puede pagar el coste o auditar las citas, el experimento no ha ganado de forma completa.

Tamaño muestral, MDE y peeking

El error más común al enseñar A/B testing es quedarse en la fórmula del efecto y no hablar de cuánta muestra hace falta para medirlo. Un experimento con 12 unidades por variante puede ser excelente para aprender el mecanismo, pero no para tomar una decisión global si queremos detectar un cambio pequeño.

El MDE (*minimum detectable effect*) es el efecto mínimo que el experimento está diseñado para detectar con una potencia determinada. La fórmula siguiente es la aproximación normal clásica para tamaño muestral en dos grupos de igual tamaño; no es una ocurrencia del capítulo, sino una pieza estándar de cálculo de potencia y tamaño muestral.[20]

$$n \approx \frac{2(z_{1-\alpha/2} + z_{1-\beta})^2 \sigma^2}{\delta^2}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
n	Unidades necesarias por variante.	`1530` para el MDE del cuaderno.
α	Probabilidad aceptada de falso positivo.	`0.05`.
$1-\beta$	Potencia: probabilidad de detectar el efecto si existe.	`0.8`.
$z_{1-\alpha/2}$	Cuantil normal para el nivel de confianza.	`1.959964`.
$z_{1-\beta}$	Cuantil normal para la potencia.	`0.841621`.
σ^2	Varianza aproximada de la métrica.	Para binaria: $p(1-p)$.
δ	MDE que queremos detectar.	`0.05`.

Para una métrica binaria como `resolved`, usamos:

$$\sigma^2 = p(1-p)$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
p	Tasa base esperada.	`0.58`.
$1-p$	Complemento de la tasa base.	`0.42`.
σ^2	Varianza binaria aproximada.	`0.2436`.

El cuaderno calcula:

PARÁMETRO	VALOR
Baseline	`0.58`
Alpha	`0.05`
Potencia	`0.8`
MDE planificado	`0.05`
n actual por variante	`12`
n recomendado por variante	`1530`
MDE aproximado con n actual	`0.564504`

Esta tabla es una de las más importantes del capítulo porque traduce una intuición a una restricción dura. Con pocas unidades puedes detectar solo efectos enormes. Eso no invalida el experimento: puede servir para validar instrumentación, confirmar que el pipeline produce artefactos y aprender si los guardrails se mueven. Pero no deberíamos venderlo como prueba definitiva de impacto global.

Para un equipo de ingeniería, el MDE cambia la conversación. En vez de preguntar “¿ha salido significativo?”, preguntas “¿qué cambio mínimo necesitábamos detectar y cuánta muestra hacía falta para eso?”. Si el negocio necesita detectar una mejora de dos puntos y tu experimento solo puede detectar veinte, el resultado negativo no demuestra ausencia de efecto; demuestra que el diseño no tenía suficiente resolución.

Eso explica por qué el capítulo deja el experimento en `review`: la señal observada es positiva, pero la muestra solo permite detectar efectos enormes. Para decidir con un MDE pequeño, necesitamos más unidades.

Peeking es mirar resultados muchas veces y decidir cuando la gráfica parece favorable. Statsig recomienda configurar sequential testing cuando se quiere evitar que mirar varias veces aumente falsos positivos.[21] En un contrato serio, la regla debe estar escrita antes:

DECISIÓN	QUÉ ESCRIBIR ANTES DE EMPEZAR
Cuándo mirar	Fecha final, número de looks o regla secuencial.
Qué métrica decide	Una primaria, no cinco primarias cambiantes.
Qué guardrails bloquean	Umbrales concretos y dirección.
Qué pasa si queda `review`	Ampliar muestra, repetir ventana o limitar rollout.
Qué no se permite	Cambiar metodología al ver el resultado.

Johari, Pekelis y Walsh explican por qué mirar repetidamente un A/B test y parar cuando conviene aumenta el riesgo de falso positivo si el análisis no lo contempla.[22] La solución profesional no es “no mires nunca”, porque producto y operación necesitan seguimiento. La solución es declarar una regla de mirada.

ESTRATEGIA	QUÉ HACE	CUÁNDO ENCAJA	RIESGO SI SE EXPLICA MAL
Mirada única	Analizas al final de la ventana.	Experimentos simples y métricas maduras.	Tardas en detectar problemas operativos.
Sequential testing	Permite mirar varias veces con control del error.	Producto necesita vigilancia durante la prueba.	Parece “barra libre” si no se fija antes.
Alpha spending	Reparte el presupuesto de error entre miradas.	Hay calendario de revisiones planificado.	La implementación debe ser la misma que el análisis.
Límites Pocock	Usa umbral parecido en varias miradas.	Quieres regla sencilla de comunicación.	Puede ser menos agresivo al principio.
Límites O'Brien-Fleming	Umbral muy estricto al principio y más flexible al final.	Quieres parar pronto solo con evidencia muy fuerte.	Es más difícil de explicar a negocio.

Pocock y O'Brien-Fleming son diseños clásicos de análisis secuencial en ensayos, pero la lección nos sirve aquí: si vas a mirar varias veces, esa política forma parte del experimento, no de la conversación posterior.[23][24]

También conviene ejecutar A/A antes de A/B. En A/A, ambas variantes se comportan igual. Si el sistema detecta efecto donde no debería, tenemos problema de instrumentación, asignación, logging o análisis. Es una prueba humilde y muy de ingeniería: antes de medir impacto, comprobamos que el contador mide.

Métricas, jerarquía y comparaciones múltiples

Otro tropiezo habitual: medir muchas cosas y quedarse con la que sale bonita. Si miras diez métricas, cinco slices y tres ventanas, alguna diferencia llamativa aparecerá por azar. Por eso el plan debe separar métrica primaria, guardrails y diagnósticas.

TIPO	DECIDE	EJEMPLO	CÓMO SE INTERPRETA
Primaria	Sí	`resolved_day_7`.	Una, fijada antes.
Guardrail	Bloquea	`latency_ms`, `cost_eur`, `citation_valid`.	No gana el experimento, pero puede impedir publicar.
Diagnóstica	Explica	`retrieval_precision`.	Ayuda a depurar, no decide sola.
Slice	Matiza	`segment=becas`.	Señal de heterogeneidad o riesgo.
Ventana secundaria	Complementa	`day_1`, `day_30`.	Ayuda a maduración, no reemplaza primaria sin plan.

La jerarquía de métricas es una forma de honestidad. Una primaria decide porque representa la apuesta principal. Los guardrails protegen lo que no quieres romper. Las diagnósticas explican mecanismos y ayudan a depurar. Los slices muestran dónde la media puede estar escondiendo daño o beneficio. Si todo decide, nada decide; si nada decide, el experimento no cambia la acción.

Si declaramos muchos resultados como “primarios”, sube el riesgo de falso descubrimiento. Una corrección conservadora es Bonferroni:

$$\alpha^* = \frac{\alpha}{m}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
α^*	Umbral corregido por número de pruebas.	`0.01` si $\alpha = 0.05$ y $m = 5$.
α	Umbral original.	`0.05`.
m	Número de pruebas consideradas.	`5` guardrails o hipótesis.

Benjamini y Hochberg propusieron controlar la tasa de descubrimientos falsos, menos conservadora que Bonferroni cuando se exploran muchas hipótesis.[25]

En este facsímil la regla pedagógica es clara:

```
Una métrica primaria.
Guardrails bloqueantes.
Slices y diagnósticas como explicación, no como celebración automática.
```

CUPED: reducir varianza sin cambiar la pregunta

CUPED usa información previa al experimento para reducir ruido. Deng, Xu, Kohavi y Walker propusieron usar datos pre-experimento para mejorar la sensibilidad de experimentos online.[26] Microsoft ExP lo describe como técnica de reducción de varianza en A/B testing.[27]

La idea no es manipular el resultado. Es restar parte del ruido explicable por una covariable medida antes del tratamiento:

$$Y_i^* = Y_i - \theta(X_i - \bar{X})$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
Y_i^*	Resultado ajustado por CUPED.	`resolved` ajustado.
Y_i	Resultado original.	`resolved = 1`.
X_i	Covariable previa al experimento.	`historical_resolution_rate`.
$\bar{X}$	Media de la covariable previa.	Media histórica del dataset.
θ	Peso del ajuste.	`3.107096` en el cuaderno.

El peso suele estimarse así:

$$\theta = \frac{Cov(Y, X)}{Var(X)}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$Cov(Y, X)$	Covarianza entre resultado y covariable previa.	Relación entre resolución y histórico.
$Var(X)$	Varianza de la covariable previa.	Dispersión de `historical_resolution_rate`.
θ	Coeficiente de ajuste.	`3.107096`.

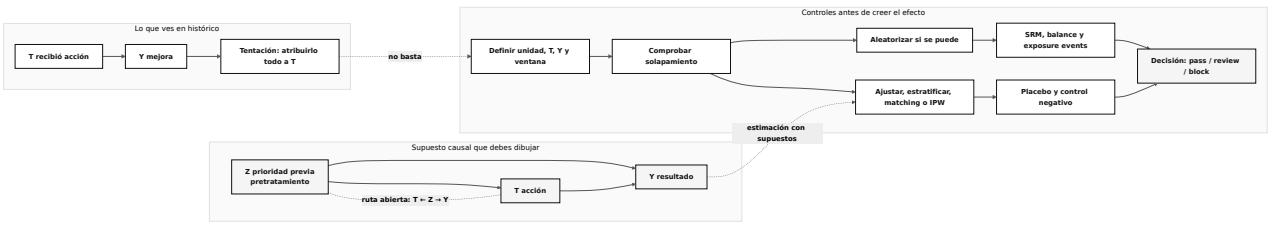
En el cuaderno, CUPED mantiene el efecto en `0.25`, pero baja el error estándar de `0.186339` a `0.162557`. Eso no convierte una señal en verdad absoluta, pero ayuda a medir con menos ruido.

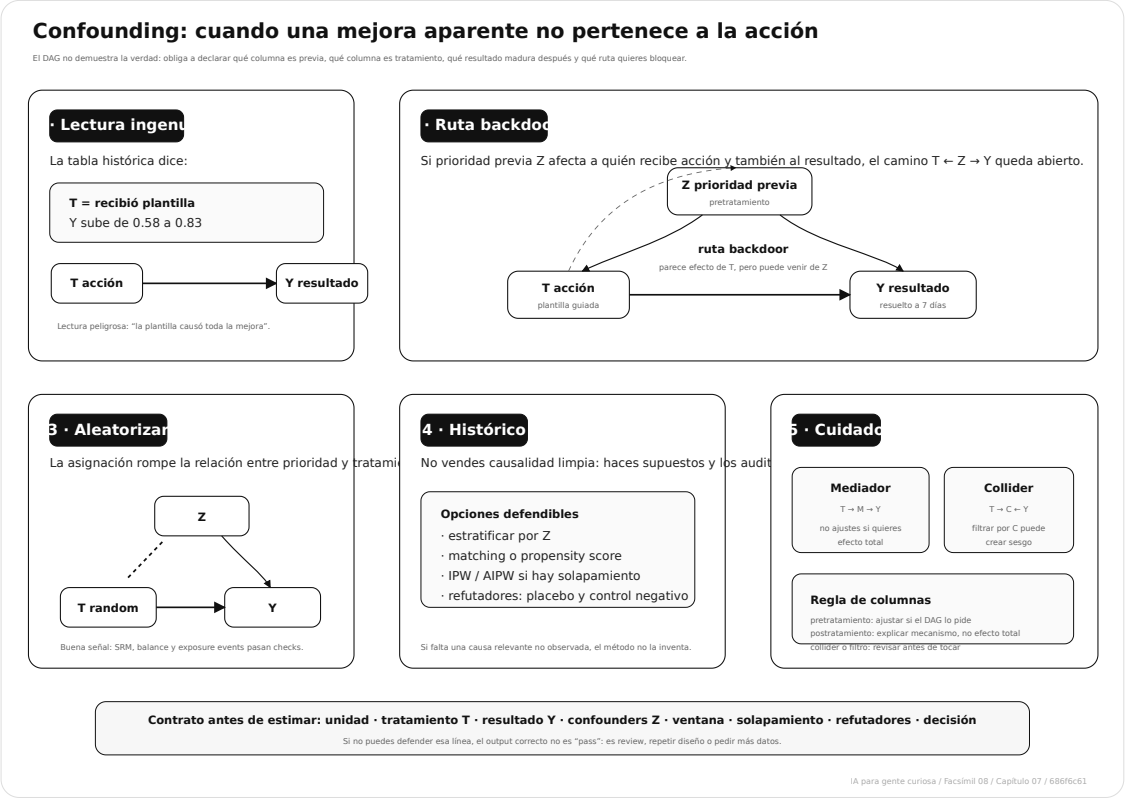
La intuición es parecida a escuchar una conversación con ruido de fondo y conocer de antemano parte de ese ruido. Si antes del experimento ya sabemos que algunos casos eran históricamente más fáciles, podemos usar esa información previa para reducir variabilidad. Pero hay una condición clave: la covariable debe estar medida antes del tratamiento. Si usas una variable afectada por la intervención, ya no estás reduciendo ruido; estás cambiando la pregunta.

Causalidad observacional: cuando no puedes aleatorizar

No siempre podemos hacer un A/B test. A veces trabajamos con datos históricos. En ese caso, el primer deber es no vender la asociación como efecto.

El problema típico se llama confounding. Una variable Z influye en el tratamiento y en el resultado:





> El confounding no es "una variable más": es una ruta alternativa que puede explicar por qué tratamiento y resultado se mueven juntos. Primero se dibuja el supuesto; luego se elige experimento, ajuste o revisión.

El dibujo tiene una consecuencia práctica muy seria: antes de estimar nada hay que decidir qué columnas pertenecen al pasado, cuáles describen la acción y cuáles son resultado. En un sistema de IA esto se vuelve fácil de ensuciar. Una prioridad calculada después de que el modelo intervenga no puede usarse como si fuese contexto previo. Una etiqueta humana generada tras ver la respuesta del asistente no es una variable inocente. Un score de riesgo que cambia durante la conversación puede ser parte del mecanismo, no una causa previa que podamos ajustar sin pensar.

Este punto suele separar un análisis defendible de un informe vistoso. Si el equipo no puede explicar por qué Z ocurre antes que T, por qué T ocurre antes que Y, y por qué la ventana de Y es suficiente para que el efecto madure, el análisis causal todavía no está listo. Dibujar el DAG no garantiza que tengas razón, pero obliga a decir dónde podría estar el error. Esa humildad técnica es exactamente lo que se necesita cuando un número va a decidir si una plantilla, un modelo o un flujo de revisión se publica.

Si no ajustamos por Z, podemos atribuir a T lo que en realidad viene de la prioridad previa. El ajuste por backdoor escribe:

$$P(Y \mid do(T = t)) = \sum_{z} P(Y \mid T = t, Z = z) P(Z = z)$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
T	Tratamiento o acción.	Recibir plantilla.
t	Valor del tratamiento.	`1` o `0`.
Y	Resultado.	Caso resuelto.
Z	Confounder observado.	Prioridad previa.
z	Nivel concreto de Z.	`high`, `medium`, `low`.
$P(Y \mid T = t, Z = z)$	Resultado dentro de un estrato comparable.	Resolución de casos `medium` con plantilla.
$P(Z = z)$	Peso del estrato en la población.	Proporción de prioridad `medium`.

Esto exige solapamiento. Si en prioridad `low` nadie recibió tratamiento, no podemos comparar tratamiento contra control dentro de ese nivel. El cuaderno lo muestra: el efecto ingenuo es `0.5`, pero al estratificar por prioridad baja a `0.0375` en la población estimable y queda en `review` por falta de solapamiento.

DoWhy separa modelado causal, identificación, estimación y refutación.[28] Su documentación actual insiste en tratar los supuestos como ciudadanos de primera clase y separar identificación de estimación.[29] EconML aplica técnicas de machine learning a estimación de respuestas causales individuales o heterogéneas en datos experimentales u observacionales.[30]

La regla para el facsímil es sencilla: si no puedes escribir supuestos, estimando, población y prueba de robustez, todavía no tienes conclusión causal.

Una herramienta clásica en observacional es el propensity score: la probabilidad de recibir tratamiento dado el contexto observado.[31] Se escribe:

$$e(x) = P(T = 1 \mid X = x)$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$e(x)$	Propensity score.	Probabilidad de recibir plantilla.
$T = 1$	Recibir tratamiento.	`received_action = 1`.
$X = x$	Contexto observado.	Prioridad, segmento, histórico.

Sirve para emparejar, ponderar o estratificar casos con probabilidades parecidas de recibir tratamiento. Pero no resuelve el sesgo por sí solo: solo ajusta por variables observadas. Si falta una causa importante, el sesgo puede seguir ahí. Imbens y Rubin desarrollan con detalle este tipo de inferencia causal basada en supuestos explícitos.[32]

Métodos observacionales que conviene reconocer

En ingeniería de IA vas a ver datasets históricos donde nadie aleatorizó nada: campañas, prompts usados por operadores, colas priorizadas, descuentos, plantillas, cambios de ranking, modelos servidos solo a ciertos clientes o herramientas activadas por segmento. En esos casos no hay magia. Hay familias de métodos, cada una con supuestos y fallos.

MÉTODO	QUÉ INTENTA HACER	SIRVE CUANDO	NO SIRVE SI
Matching	Comparar casos tratados con controles parecidos.	Tienes covariables observadas ricas y buen solapamiento.	Hay variables importantes no medidas o no hay controles comparables.
IPW	Ponderar por la probabilidad inversa de recibir tratamiento.	Quieres reconstruir una población más comparable con propensity score.	Hay propensity cercano a 0 o 1; los pesos explotan.
Estimador doblemente robusto	Combinar modelo de resultado y modelo de tratamiento.	Puedes modelar resultado y asignación con cierto cuidado.	Ambos modelos están mal o faltan variables clave.
Difference-in-differences	Comparar cambios antes/después entre tratado y control.	Hay una intervención temporal y tendencia paralela plausible.	Los grupos ya venían cambiando distinto antes de la intervención.
Regression discontinuity	Mirar unidades cerca de un umbral de asignación.	La regla de asignación tiene corte claro.	La gente manipula el umbral o no hay suficientes casos cerca.
Variable instrumental	Usar una variable que empuja el tratamiento sin afectar directamente al resultado.	El instrumento es defendible y fuerte.	El instrumento afecta al resultado por otro camino.

Estos métodos no son una escalera donde uno sea “mejor” que el anterior. Son respuestas distintas a problemas distintos. Matching intenta construir comparaciones parecidas. IPW intenta reponderar una población que no fue asignada al azar. Difference-in-differences aprovecha cambios temporales. Regression discontinuity explota una regla de corte. Variables instrumentales buscan una fuente externa de variación. El error de ingeniería es elegir el método por moda o por disponibilidad de librería, no por el mecanismo que produjo los datos.

Para un alumno de datos, la pregunta útil es: ¿qué historia tendría que ser cierta para que este método fuese razonable? Si digo matching, necesito creer que mis covariables observadas capturan las diferencias relevantes. Si digo difference-in-differences, necesito mirar tendencias previas. Si digo instrumento, necesito defender que el instrumento no afecta al resultado por otro camino. Esta parte no se resuelve con más código; se resuelve con linaje, conocimiento de negocio y una discusión honesta de supuestos.

Una formulación clásica de ponderación por probabilidad inversa, emparentada con el estimador de Horvitz-Thompson, estima el efecto medio así:[33]

$$\widehat{ATE}_{IPW} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{T_i Y_i}{e(X_i)} - \frac{(1-T_i)Y_i}{1-e(X_i)}\right)$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$\widehat{ATE}_{IPW}$	Efecto medio estimado con ponderación inversa.	Efecto de enviar plantilla ajustando por probabilidad de recibirla.
T_i	Indicador de tratamiento.	`1` si recibió plantilla.
Y_i	Resultado observado.	`resolved`.
$e(X_i)$	Propensity score.	Probabilidad estimada de recibir plantilla dado el contexto.
n	Número de unidades.	Casos históricos observados.

En palabras: si un caso tenía poca probabilidad de recibir tratamiento y aun así lo recibió, pesa más porque representa situaciones raras dentro del histórico. El riesgo es que pesos muy grandes conviertan unas pocas filas en dueñas de la conclusión. Por eso el solapamiento no es un detalle: es una condición de trabajo.

Los estimadores doblemente robustos combinan un modelo de resultado $\hat{\mu}_t(X)$ con un propensity score $\hat{e}(X)$. Bang y Robins los desarrollan en el contexto de datos faltantes e inferencia causal.[34] Una forma habitual del estimador AIPW es:

$$\hat{\tau}_{AIPW} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{\mu}_1(X_i) - \hat{\mu}_0(X_i) + \frac{T_i(Y_i - \hat{\mu}_1(X_i))}{\hat{e}(X_i)} - \frac{(1-T_i)(Y_i - \hat{\mu}_0(X_i))}{1-\hat{e}(X_i)}\right]$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$\hat{\tau}_{AIPW}$	Efecto estimado con ajuste doblemente robusto.	Impacto de una política sobre resolución.
$\hat{\mu}_1(X_i)$	Resultado esperado si el caso i recibe tratamiento.	Resolución esperada con plantilla.
$\hat{\mu}_0(X_i)$	Resultado esperado si el caso i queda en control.	Resolución esperada sin plantilla.
$\hat{e}(X_i)$	Propensity score estimado.	Probabilidad de recibir plantilla.
T_i, Y_i	Tratamiento y resultado observados.	Acción y resolución en el dataset.

En palabras: el estimador intenta corregir dos cosas a la vez: cómo se asignó el tratamiento y qué resultado esperaba cada tipo de caso. No autoriza a apagar el criterio. Si los datos no tienen las variables que explican la asignación real, el estimador puede seguir siendo elegante y equivocado.

Cuando existe un antes y un después, *difference-in-differences* compara cambios, no niveles. Card y Krueger lo usaron en un estudio clásico sobre salario mínimo y empleo.[35] La forma básica es:

$$\hat{\delta}_{DID} = (\bar{Y}_{T,post} - \bar{Y}_{T,pre}) - (\bar{Y}_{C,post} - \bar{Y}_{C,pre})$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$\hat{\delta}_{DID}$	Estimación difference-in-differences.	Cambio atribuible a publicar una nueva política.
$\bar{Y}_{T,post}$, $\bar{Y}_{T,pre}$	Resultado medio del grupo tratado después y antes.	Resolución tras activar plantilla y antes.
$\bar{Y}_{C,post}$, $\bar{Y}_{C,pre}$	Resultado medio del grupo control después y antes.	Resolución en grupo sin plantilla.

En palabras: si ambos grupos venían evolucionando igual antes del cambio, la diferencia de cambios puede aproximar el efecto. Si las tendencias previas ya eran distintas, el método puede convertir una tendencia natural en causalidad falsa.

Regression discontinuity se usa cuando una regla asigna tratamiento al cruzar un umbral: por ejemplo, “si el score de riesgo supera 0.80, entra revisión humana”. Lee y Lemieux resumen este diseño en econometría aplicada.[36] La lectura ingenieril es comparar casos muy cerca del

corte: un caso con 0.799 y otro con 0.801 deberían parecerse mucho salvo por la regla. Si el score se manipula, si el umbral cambia sin versionado o si hay pocos casos cerca, la conclusión se debilita.

Las variables instrumentales entran cuando hay una variable Z que cambia la probabilidad de tratamiento, pero no debería afectar al resultado salvo a través de ese tratamiento. Angrist, Imbens y Rubin formalizaron la identificación de efectos causales con instrumentos.[37] En el caso simple, el estimador de Wald se escribe:

$$\hat{\beta}_{IV} = \frac{Cov(Z, Y)}{Cov(Z, T)}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$\hat{\beta}_{IV}$	Efecto estimado mediante instrumento.	Impacto de recibir revisión humana inducida por una regla externa.
Z	Instrumento.	Disponibilidad aleatoria de un equipo revisor.
T	Tratamiento.	Recibir revisión humana.
Y	Resultado.	Resolver correctamente.
$Cov(\cdot, \cdot)$	Covarianza entre variables.	Asociación del instrumento con tratamiento y resultado.

En palabras: el instrumento tiene que mover el tratamiento, pero no puede tener otro camino hacia el resultado. Es una condición fuerte. En un sistema de IA, muchas variables que parecen instrumentos no lo son: turno, canal, país o proveedor suelen afectar otras cosas además del tratamiento.

Refutar antes de creerse el número

DoWhy llama *refuters* a pruebas que intentan cuestionar una estimación: añadir una causa común aleatoria, usar un tratamiento placebo, mirar subconjuntos o probar sensibilidad.[38] El cuaderno añade una versión mínima y ejecutable de esa idea. No pretende competir con DoWhy, EconML o CausalML; pretende que el gesto quede claro.

REFUTADOR DEL CUADERNO	QUÉ PREGUNTA	QUÉ HARÍA SALTAR UNA ALERTA
Tratamiento placebo desplazado	¿Aparece efecto incluso con una acción falsa?	Un efecto placebo grande.
Variable previa como control negativo	¿La acción ya estaba asociada a algo anterior?	Diferencias fuertes en `historical_resolution_rate`.
Estabilidad por segmento	¿El efecto cambia demasiado por slice?	Un promedio global que esconde heterogeneidad.
Solapamiento por prioridad	¿Hay tratamiento y control en cada estrato?	Estratos con solo acción o solo control.

Los refutadores son una forma sana de trabajar contra nuestras ganas de tener razón. En un proyecto real siempre hay presión para convertir una señal prometedora en lanzamiento, sobre todo si la variante coincide con lo que el equipo quería probar. Un placebo, un control negativo o una revisión por segmento no son trabas académicas: son ensayos de resistencia. Si una conclusión causal es frágil, conviene descubrirlo antes de meterla en una política de producto, una cola de soporte o una decisión automatizada.

El mensaje para ingeniería es sobrio: una estimación observacional debe sobrevivir a intentos razonables de refutación antes de convertirse en decisión. Si se rompe al primer placebo, no es una derrota del equipo; es una victoria del sistema de revisión.

Uplift y CATE: actuar donde cambia algo

El ATE puede ser positivo y aun así esconder una decisión mala. Quizá la intervención no ayuda a todos por igual. En IA aplicada, muchas veces queremos saber dónde cambia algo.

El efecto condicionado se escribe:

$$CATE(x) = E[Y(1) - Y(0) \mid X = x]$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$CATE(x)$	Efecto medio condicionado al contexto x.	Efecto en `segment=becas`.
$X = x$	Contexto o características del caso.	Segmento, idioma, canal, prioridad.
$Y(1)$	Resultado con tratamiento.	Resolución con plantilla.
$Y(0)$	Resultado sin tratamiento.	Resolución sin plantilla.

Esto conecta directamente con slices. En el cuaderno:

SEGMENTO	CONTROL	TRATAMIENTO	EFECTO OBSERVADO
`becas`	`0.25`	`0.75`	`0.50`
`matricula`	`1.00`	`1.00`	`0.00`
`practicas`	`0.50`	`0.75`	`0.25`

Esta lectura cambia el tipo de producto que construirías. Si el efecto vive sobre todo en `becas`, quizá no necesitas encarecer todo el flujo con una plantilla guiada, un modelo más grande o una revisión humana universal. Quizá necesitas una política específica para ese segmento, una evaluación adicional en ese slice y una explicación de por qué ahí sí aparece margen de mejora. CATE y uplift sirven para pasar de “la media sube” a “esta acción merece aplicarse aquí, y no necesariamente allí”.

La decisión no debería ser “plantilla para todo el mundo”. Podría ser: ampliar muestra, estudiar por qué `becas` concentra mejora y comprobar que `matricula` no necesita gasto adicional.

Del experimento al rollout

Un resultado experimental no es el final del trabajo. Es la entrada a una decisión de release. En sistemas de IA, publicar al 100% sin escalones puede mezclar tres problemas a la vez: efecto causal, estabilidad operativa y coste.

Una política de rollout sobria puede ser:

PASO	QUÉ OCURRE	QUÉ SE MIRA
A/A	Variantes iguales.	SRM, exposición, métricas estables.
A/B pequeño	Primera ventana controlada.	Señal, guardrails, slices.
Ramp 5%	Tráfico real limitado.	Latencia, coste, errores y trazas.
Ramp 25%	Más diversidad de casos.	CATE, segmentos raros, operación humana.
Ramp 50%	Estrés realista.	SLOs y presupuesto de error.
100%	Publicación completa.	Monitorización post-release y rollback.

Un rollout no es un premio por haber ganado una tabla. Es otra fase del experimento, con otra clase de riesgos. En tráfico pequeño sueles ver casos limpios; al crecer aparecen idiomas raros, documentos incompletos, picos de latencia, límites de proveedor, operadores saturados y segmentos que no estaban representados. Por eso el rollout tiene que conservar la misma disciplina del experimento: versión, exposición, métricas, guardrails y decisión escrita.

En el contrato del cuaderno aparece:

```
{
  "initial_ramp_percent": 5,
  "ramp_steps_percent": [5, 25, 50, 100],
  "rollback_if_guardrail_blocks": true,
  "publish_requires_status": "pass"
}
```

Esto no es decoración. Si el experimento queda en `review`, no debería pasar a 100%. Puede pasar a otra ventana de medición, a un A/A pendiente, a un rollout limitado o a una mejora de instrumentación. La salida profesional no es “ganó B”. La salida profesional es: **qué hacemos ahora, con qué riesgo y qué evidencia queda guardada**.

Bandits y experimentos adaptativos

Un A/B test clásico mantiene proporciones estables mientras mide. Eso es útil porque simplifica la interpretación: si control y tratamiento están bien asignados, estimar el efecto es directo. Pero a veces el objetivo no es solo medir; también queremos aprender y asignar más tráfico a variantes que parecen mejores durante el proceso.

Ahí aparecen los bandits. En un problema de bandits elegimos acciones, observamos recompensa y vamos actualizando la política de asignación. Sutton y Barto lo explican como una tensión entre exploración y explotación: probar para aprender frente a usar lo que parece mejor.[39]

La idea mínima se puede escribir así:

$$a_t = \arg\max_a Q_t(a)$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
a_t	Acción elegida en el paso t.	Variante de prompt para el ticket actual.
a	Acción candidata.	`control`, `treatment_a`, `treatment_b`.
$Q_t(a)$	Valor estimado de la acción en el paso t.	Tasa media de resolución observada.
$\arg\max$	Acción con mayor valor estimado.	Variante que parece mejor.

Pero si siempre elegimos la que parece mejor, podemos dejar de explorar demasiado pronto. Una regla epsilon-greedy añade exploración:

$$a_t = \begin{cases} \textbf{acción aleatoria} & \textbf{con probabilidad } \epsilon \\ \arg\max_a Q_t(a) & \textbf{con} \end{cases}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
ϵ	Probabilidad de explorar.	`0.1`.
$1 - \epsilon$	Probabilidad de explotar lo mejor conocido.	`0.9`.
$Q_t(a)$	Estimación acumulada por acción.	Calidad media por variante.

Para IA aplicada, la tabla de decisión sería:

DISEÑO	ÚTIL CUANDO	NO LO USES SI
A/B fijo	Necesitas una conclusión clara y auditable.	La pérdida de oportunidad durante la prueba es muy alta.
A/A	Quieres validar instrumentación.	Esperas medir efecto real.
Bandit	Quieres aprender y reasignar tráfico mientras operas.	Necesitas estimación simple del ATE con reparto estable.
Rollout progresivo	Quieres publicar con control operativo.	Lo confundes con experimento causal completo.
Holdout persistente	Quieres medir impacto largo plazo.	No puedes mantener grupo sin tratamiento.

La decisión entre A/B, bandit, rollout y holdout depende de qué te duele más: ignorancia, coste de oportunidad, riesgo operativo o pérdida de una referencia estable. Si estás probando un prompt de bajo riesgo y el volumen es alto, quizá un A/B fijo te da una lectura limpia. Si una variante mala puede dañar a usuarios o consumir mucho coste, un rollout progresivo tiene más sentido. Si el sistema aprende mientras sirve, un bandit puede reducir exposición a malas variantes, pero a cambio exige más cuidado al analizar. No hay una opción profesional por defecto; hay una opción coherente con el daño posible.

Los bandits pueden ser muy útiles en recomendación, prompts alternativos o selección de respuesta, pero complican inferencia, métricas tardías y comunicación. Si cambias la asignación durante el experimento, el análisis debe saberlo. No puedes tratarlo luego como si hubiese sido un A/B fijo.

De quién viene cada fórmula y por qué aparece aquí

Este capítulo usa fórmulas porque el tema las necesita. No están para decorar ni para dar una apariencia científica. Cada una corresponde a una familia reconocible de la literatura de inferencia causal, experimentación online, estadística aplicada o aprendizaje por refuerzo. La pregunta editorial es siempre la misma: ¿esta fórmula ayuda a tomar una decisión de ingeniería más limpia?

FÓRMULA O FAMILIA	PROCEDENCIA	QUÉ APORTA AL CAPÍTULO	DECISIÓN QUE AYUDA A TOMAR
$P(Y \mid X = x)$ frente a $P(Y \mid do(X = x))$	Pearl y el marco de intervención con operador `do`.	Separa asociación observada de intervención diseñada.	No usar una correlación histórica como prueba de impacto.
$Y_i(1)$, $Y_i(0)$ y $ATE = E[Y(1) - Y(0)]$	Resultados potenciales de Rubin y formulación causal de Holland.	Explica el problema contrafactual: solo vemos uno de los dos mundos.	Diseñar control y tratamiento antes de hablar de efecto.
$\widehat{ATE} = \bar{Y}_T - \bar{Y}_C$	Estimación clásica de diferencia de medias en experimentos aleatorizados; se apoya en el marco de resultados potenciales de Rubin/Holland cuando la asignación es aleatoria.	Convierte la comparación de variantes en una magnitud interpretable.	Saber si el tratamiento mueve la métrica primaria y cuánto.
$SE(\widehat{ATE})$ e $IC_{95\%}$	Inferencia estadística clásica para diferencia de medias, como se enseña en textos de inferencia estadística.	Añade incertidumbre al efecto observado.	Evitar publicar por una señal grande pero imprecisa.
$\widehat{ATE}_{cluster}$	Ensayos y diseños con aleatorización por clúster, tratados de forma específica por Donner y Klar.	Recuerda que no todas las unidades son independientes.	Cambiar la unidad de asignación cuando hay operador, cola o equipo compartido.
$DEFF = 1 + (m - 1)\rho$ y $n_{eff} \approx n / DEFF$	Muestreo por clúster de Kish.	Traduce dependencia dentro de clúster a tamaño efectivo.	No contar eventos correlacionados como si fueran unidades independientes.
$\chi^2 = \sum_g (O_g - E_g)^2 / E_g$	Test chi-cuadrado usado para detectar desajustes de reparto, incluyendo SRM.	Comprueba si el reparto observado coincide con el esperado.	No analizar un experimento si la aleatorización o el tracking parecen rotos.
$SMD = (\bar{X}_T - \bar{X}_C) / s_{pooled}$	Diferencia de medias estandarizada usada como diagnóstico de balance; Austin la discute en el contexto de propensity score.	Permite comparar grupos antes del tratamiento en una escala común.	Detectar si control y tratamiento ya eran distintos antes de actuar.

FÓRMULA O FAMILIA	PROCEDENCIA	QUÉ APORTA AL CAPÍTULO	DECISIÓN QUE AYUDA A TOMAR
Fórmula aproximada de n para MDE	Cálculo clásico de tamaño muestral y potencia para dos grupos con aproximación normal.	Conecta muestra disponible con efecto detectable.	Distinguir “sirve para aprender” de “sirve para publicar”.
$\alpha^* = \alpha/m$	Corrección de Bonferroni para comparaciones múltiples.	Enseña el coste de mirar muchas métricas como si todas fueran primarias.	Separar métrica primaria, guardrails y diagnósticas.
$Y_i^* = Y_i - \theta(X_i - \bar{X})$ y $\theta = Cov(Y, X)/Var(X)$	CUPED de Deng, Xu, Kohavi y Walker.	Reduce varianza usando una covariable previa al experimento.	Ganar sensibilidad sin cambiar la pregunta causal.
Ajuste por backdoor	Pearl y grafos causales.	Muestra cómo ajustar por un confounder observado cuando no hay experimento.	No vender datos observacionales como causalidad si faltan supuestos o solapamiento.
$e(x) = P(T = 1 \mid X = x)$	Propensity score de Rosenbaum y Rubin.	Resume la probabilidad de recibir tratamiento dado el contexto observado.	Emparejar, ponderar o estratificar sin olvidar que solo ajusta variables observadas.
$\widehat{ATE}_{IPW}$	Ponderación por probabilidad inversa basada en Horvitz-Thompson y propensity score.	Repondera casos según lo raro que era observar su tratamiento.	Ver si el efecto depende de pesos inestables o falta de solapamiento.
$\hat{\tau}_{AIPW}$	Estimación doblemente robusta de Bang y Robins.	Combina modelo de resultado y modelo de asignación.	No depender de una sola familia de modelo cuando se trabaja con histórico.
$\hat{\delta}_{DID}$	Difference-in-differences usado en cuasi-experimentos.	Compara cambios antes/después entre tratado y control.	Evaluar cambios temporales cuando la aleatorización no existe.
$\hat{\beta}_{IV} = Cov(Z, Y)/Cov(Z, T)$	Variables instrumentales de Angrist, Imbens y Rubin en el caso simple.	Usa una fuente externa de	No aceptar un supuesto instrumental débil

FÓRMULA O FAMILIA	PROCEDENCIA	QUÉ APORTA AL CAPÍTULO	DECISIÓN QUE AYUDA A TOMAR
		variación en el tratamiento.	sin discutir exclusión y fuerza.
$CATE(x) = E[Y(1) - Y(0) \mid X = x]$	Efectos heterogéneos de tratamiento en inferencia causal moderna.	Lleva el análisis desde el promedio hacia segmentos accionables.	Decidir si publicar para todos o solo para un slice con evidencia.
$a_t = \arg\max_a Q_t(a)$ y epsilon-greedy	Bandits en aprendizaje por refuerzo, como en Sutton y Barto.	Explica exploración frente a explotación.	Elegir entre A/B fijo, bandit o rollout progresivo según la necesidad de medir o actuar.

La lectura práctica es esta: si una fórmula no cambia qué se diseña, qué se mide, qué se bloquea o qué se publica, sobra. Si cambia una decisión, entonces debe venir con fuente, símbolos, ejemplo y una traducción a operación.

Herramientas y decisiones de compra o montaje

No hace falta que cada equipo construya Microsoft ExP desde cero. Pero sí debe saber qué está comprando o montando.

OPCIÓN	ENCAJA CUANDO	PREGUNTAS ANTES DE USARLA
Feature flags simples	Solo necesitas activar o desactivar variantes.	¿Registra exposición? ¿Persistencia? ¿Auditoría?
Plataforma de experimentación	Quieres análisis, métricas y decisiones recurrentes.	¿SRM, A/A, CUPED, sequential testing, slices?
Warehouse-native	Ya tienes métricas en BigQuery, Snowflake, Databricks o similar.	¿La definición SQL de métricas es versionable?
Implementación propia	Necesitas control completo o restricciones internas.	¿Quién mantiene asignación, logs, análisis y UI?
OpenFeature + proveedor	Quieres no acoplar código a una herramienta.	¿El proveedor soporta contexto, tracking y trazas?

La herramienta correcta depende de qué pieza te falta. Si el problema es que nadie sabe qué variante recibió cada usuario, necesitas mejorar asignación y exposición, no comprar un dashboard bonito. Si el problema es que las métricas viven en el warehouse y cambian cada semana, necesitas contratos de métricas, linaje y revisión SQL. Si el problema es que producto lanza sin evidencia, necesitas release gates. Y si el problema es que el equipo no distingue asociación de causalidad, la plataforma no lo arregla sola.

Lo que no conviene comprar es “confianza”. La confianza se construye con eventos completos, métricas versionadas, checks de SRM, balance, ventanas de maduración, guardrails y revisiones. Una plataforma puede ayudarte a ejecutar esa disciplina con menos fricción, pero no puede decidir por ti si el tratamiento estaba bien definido o si el resultado medido era el que importaba.

En causalidad observacional, la elección de herramienta también importa. DoWhy te fuerza a separar modelo causal, identificación, estimación y refutación. EconML se centra más en estimar efectos heterogéneos con aprendizaje automático. CausalML ofrece métodos de uplift y CATE desde una interfaz Python orientada a casos experimentales u observacionales.[40] PyMC puede servir cuando quieres modelar incertidumbre de forma bayesiana y trabajar con propensity scores o modelos causales probabilísticos.[41]

HERRAMIENTA	MEJOR USO EN ESTE CAPÍTULO	QUÉ NO DEBES DELEGARLE
DoWhy	Escribir DAG, identificar estimando, estimar y ejecutar refutadores.	La validez de tus supuestos causales.
EconML	CATE, uplift y tratamiento heterogéneo con modelos de ML.	La definición de población, tratamiento y resultado.
CausalML	Uplift modeling, meta-learners, árboles uplift y validación práctica.	La auditoría de solapamiento y sesgo no medido.
PyMC	Causalidad con incertidumbre bayesiana y modelos probabilísticos explícitos.	Convertir un mal DAG en una buena conclusión.
GrowthBook, Statsig, LaunchDarkly	Experimentos online, flags, métricas y decisiones de producto.	La semántica de tus métricas o tus guardrails.
Eppo	Feature flags y experimentación conectada al stack de datos existente.	Que tu warehouse tenga métricas bien definidas y contratos de exposición.
PostHog	Experimentos, analítica de producto y flags en equipos que quieren empezar rápido.	Diseñar una pregunta causal limpia o una política de peeking.
Optimizely Feature Experimentation	Experimentación con SDKs y APIs en entornos más enterprise o multi-superficie.	La calidad de tus eventos, tus métricas y tu decisión de rollout.
Evidently	Monitorizar drift, calidad y comportamiento de datos/modelos alrededor del experimento.	Estimar causalidad o sustituir un A/B test.
OpenFeature	Desacoplar código de proveedor de flags.	El análisis estadístico posterior.

Leída así, la tabla no es una lista de logos. Es un mapa de responsabilidades. DoWhy o EconML viven más cerca de la inferencia causal. GrowthBook, Statsig, LaunchDarkly, Eppo, PostHog y Optimizely viven más cerca de flags, producto y análisis recurrente. Evidently ayuda alrededor del ciclo, cuando quieres saber si los datos o el comportamiento del modelo se están moviendo. OpenFeature reduce acoplamiento en código. En un equipo serio estas piezas pueden convivir, pero cada una debe tener un papel claro.

Para IA, añade estas preguntas:

PREGUNTA	POR QUÉ IMPORTA
¿La variante incluye versión de modelo, prompt y parámetros?	Cambiar `temperature`, prompt o modelo cambia el tratamiento.
¿La exposición registra fallback?	Si el modelo nuevo falla y se usa otro, no puedes contar esa unidad igual.
¿Las métricas maduran tarde?	Una resolución puede medirse días después de la exposición.
¿Hay trazas de herramienta o RAG?	Sin ellas no sabes qué contexto o tool produjo el resultado.
¿El coste se mide por unidad asignada o expuesta?	Una variante puede resolver más y aun así no compensar.

Experimentos RAG y métricas tardías

Un experimento RAG no mide solo “la respuesta gustó”. Cambiar el retriever, el reranker o el chunking cambia qué evidencia llega al modelo. Por eso las métricas deben separar resultado, calidad de recuperación y guardrails técnicos.

En el cuaderno añadimos `rag_experiment_events.csv` con estas señales:

MÉTRICA	QUÉ MIDE	POR QUÉ IMPORTA
`answer_accepted`	Si la respuesta fue aceptada.	Métrica de utilidad.
`citation_valid`	Si la cita apuntaba a evidencia válida.	Guardrail documental.
`retrieval_precision`	Si los documentos recuperados eran relevantes.	Calidad del sistema RAG, no solo del LLM.
`latency_ms`	Tiempo de respuesta.	Un reranker puede mejorar calidad y empeorar experiencia.
`cost_eur`	Coste por unidad.	La mejora puede no compensar si el coste sube demasiado.

Un experimento RAG suele fallar cuando mezcla niveles. Si cambias el chunking, estás tocando recuperación; si cambias el prompt, estás tocando generación; si cambias el reranker, estás tocando orden de evidencia; si cambias el modelo, estás tocando lectura y

síntesis. Si todo cambia a la vez y solo miras aceptación final, quizá ganes, pero no sabrás qué aprendiste. Para ingeniería, aprender qué componente movió la métrica importa casi tanto como ganar el experimento.

La salida del cuaderno deja el experimento RAG en `review`: el tratamiento mejora aceptación y recuperación, pero sube latencia y coste. Esa es una decisión realista. En IA aplicada, muchas mejoras son tradeoffs, no victorias limpias.

También añadimos `late_metric_events.csv`. Algunas métricas maduran tarde. Un caso puede parecer resuelto en el día 1 y reabrirse en el día 7. O puede necesitar seguimiento al principio y terminar bien después. Por eso la tabla de métricas debe declarar ventana:

VENTANA	QUÉ PUEDE MEDIR	RIESGO
`day_0`	Exposición, latencia, coste.	Todavía no sabes si ayudó.
`day_1`	Resolución temprana.	Puede ignorar reaperturas.
`day_7`	Resolución más estable.	Llega tarde para decisiones rápidas.
`day_30`	Retención o satisfacción persistente.	Mezcla más cambios del entorno.

La ventana es parte del contrato, no un parámetro menor. En sistemas de soporte, una respuesta puede parecer buena en el momento y generar una reapertura dos días después. En sistemas de recomendación, una acción puede subir clics inmediatos y dañar satisfacción posterior. En un asistente con herramientas, una ejecución rápida puede ocultar una corrección manual posterior. Si la métrica madura tarde, el sistema de decisión tiene que esperar o declarar que decide con una métrica temprana y limitada.

Un sistema profesional no pregunta solo “¿qué métrica?”. Pregunta “¿en qué ventana, con qué unidad y con qué maduración?”.

Schema, CI y contrato de análisis

El cuaderno incluye `warehouse_schema.sql` para dejar una idea de arquitectura mínima. Hay cuatro tablas:

TABLA	QUÉ GUARDA	POR QUÉ EXISTE
`experiment_units`	Unidad, variante, flag y contexto.	Saber qué se asignó.
`exposure_events`	Exposición real a la variante.	Saber qué se vio o recibió.
`metric_events`	Métricas por unidad y ventana.	Separar exposición de resultado.
`experiment_decisions`	Decisión final y evidencia.	Versionar la salida profesional.

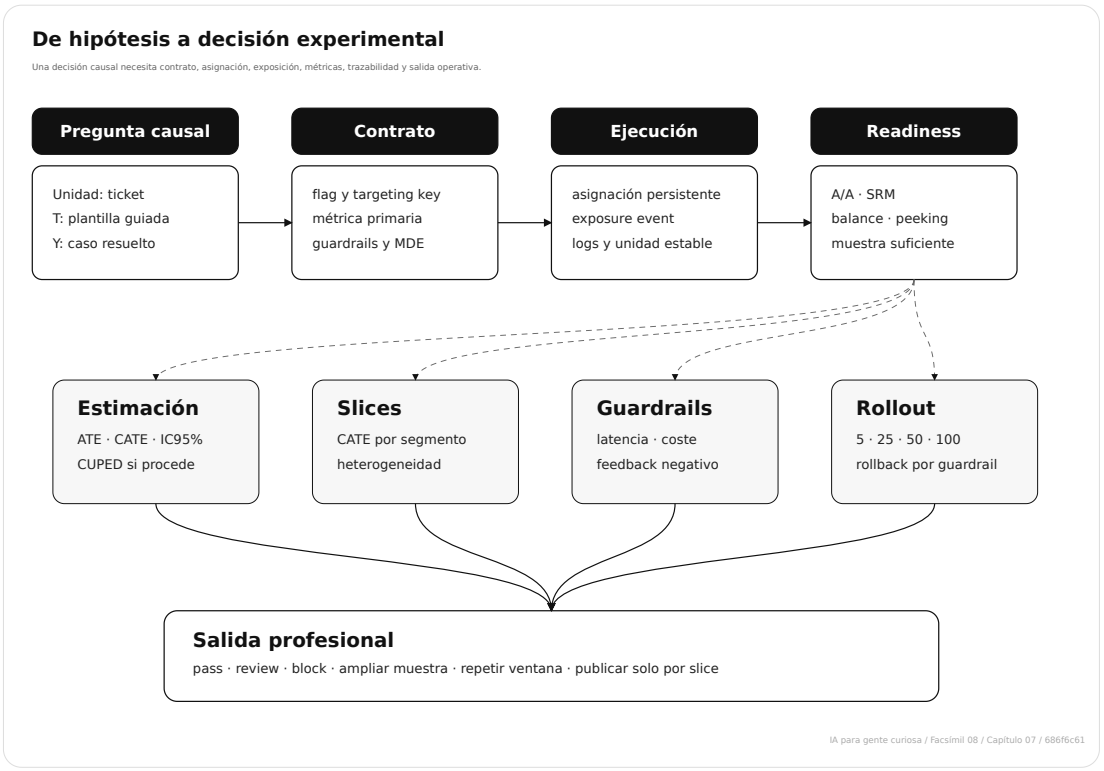
Este schema no es una propuesta universal de base de datos. Es una forma mínima de obligar a que el experimento deje rastro. La unidad asignada, la exposición real y la métrica madura no deberían vivir mezcladas en una misma fila improvisada. Separarlas evita errores muy comunes: contar unidades no expuestas, medir antes de tiempo, perder la versión de la flag o no poder reconstruir por qué una release pasó.

También permite que el CI revise algo más que “el script corre”. Puede comprobar que el análisis produjo un reporte, que los campos críticos existen, que la decisión está escrita y que los guardrails no se han saltado. Esta es la diferencia entre un notebook útil para explorar y un sistema de ingeniería capaz de sostener decisiones repetidas.

Además, `ci_experiment_gate.py` revisa que existan contratos, outputs y campos obligatorios del reporte. No bloquea por estar en `review`; lo deja explícito. Bloquearía si faltan archivos, si el schema falla, si SRM queda en `block` o si algún guardrail bloquea.

Eso es importante para equipos de ingeniería: una decisión en `review` puede ser aceptable si significa “seguir midiendo”. Una decisión en `block` significa “no publiques, hay una condición rota”. El CI no debe sustituir criterio, pero sí debe impedir que un experimento incompleto parezca listo.

Arquitectura de una decisión experimental



> Una decisión experimental no empieza en el p-value: empieza en la pregunta causal y termina en una salida operativa versionada.

En el día a día

En producción no basta con decir que el tratamiento “gana”. El resultado tiene que pasar por un contrato.

En el cuaderno del facsímil, el A/B test tiene 24 unidades: 12 en control y 12 en tratamiento. La métrica primaria es `resolved`, que debe subir. Los guardrails vigilan feedback negativo, latencia y coste.

SEÑAL	RESULTADO	LECTURA
Control `resolved`	`0.583333`	Línea base.
Treatment `resolved`	`0.833333`	Mejora observada.
ATE observado	`0.25`	Señal positiva.
IC95%	`[-0.115224, 0.615224]`	Intervalo demasiado ancho.
SRM	`pass`	Asignación 50/50 correcta.
Balance previo	`pass`	Covariables iniciales equilibradas.
Guardrails	`pass`	Coste y latencia no bloquean.
Decisión	`review`	Prometedor, pero falta precisión.

El readiness añade otra capa:

SEÑAL DE READINESS	RESULTADO	LECTURA
A/A previo	`review`	Falta documentarlo antes de confiar en el A/B.
Exposure event	`pass`	El contrato exige `experiment_exposure`.
Asignación persistente	`pass`	La unidad conserva variante.
n recomendado por variante	`1530`	El MDE de `0.05` exige mucha más muestra.
MDE aproximado con n actual	`0.564504`	Con 12 por variante solo detectas efectos enormes.
Política de peeking	`pass`	Hay una sola mirada planificada.
Rollout	`pass`	Hay pasos 5/25/50/100 y rollback por guardrail.

El análisis observacional cuenta otra historia:

LECTURA	RESULTADO
Efecto ingenuo	`0.5`
Efecto estratificado por prioridad	`0.0375`
Población estimable por prioridad	`0.75`
Decisión	`review`

La diferencia enseña la lección del capítulo: lo que parece enorme en datos históricos puede encogerse cuando comparas contextos más parecidos.

Por qué debería importarte

Si no separas predicción de intervención, puedes construir sistemas que gastan recursos donde no cambian nada. Puedes enviar acciones a quienes ya iban a resolver. Puedes castigar un segmento porque aparece asociado a un resultado, cuando en realidad recibe casos más difíciles. Puedes publicar un cambio por una métrica primaria bonita mientras se degrada latencia, coste o experiencia.

Para ingeniería de IA, causalidad no es lujo académico. Es control de daños en decisiones que actúan sobre el mundo.

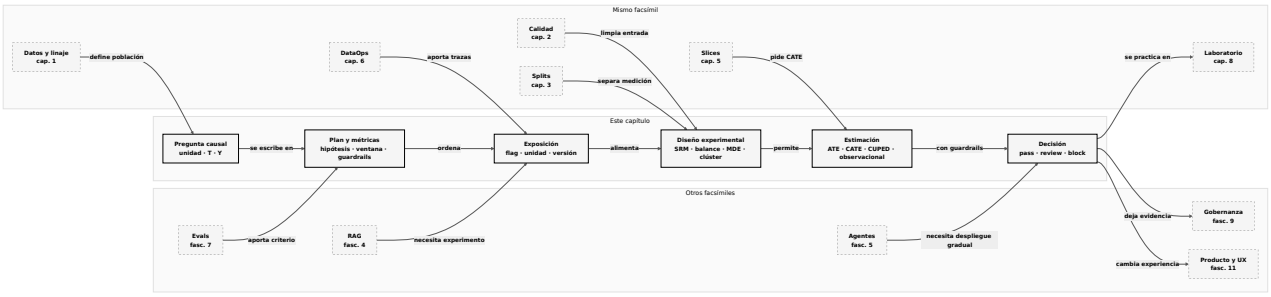
Dónde solía tropezar yo

TROPIEZO	POR QUÉ OCURRE	ANTÍDOTO
Usar predicción como efecto	El score parece accionable.	Preguntar si queremos $P(Y \mid X)$ o $P(Y \mid do(X))$.
Celebrar un ATE sin guardrails	La métrica primaria sube.	Revisar coste, latencia, feedback y slices antes de decidir.
Mirar el resultado antes de validar asignación	Queremos saber quién ganó.	Comprobar SRM, balance y exposición antes del efecto.
Vender datos históricos como experimento	Hay muchas filas y parece serio.	Escribir DAG, confounders, solapamiento y supuestos.
Ignorar heterogeneidad	El promedio es cómodo.	Revisar CATE o slices antes de publicar una política global.
Parar cuando el resultado gusta	La señal temprana seduce.	Fijar criterio de parada y decisión antes de mirar.
No registrar exposición real	La asignación parece suficiente.	Medir solo unidades que vieron o recibieron la variante.
No calcular MDE	El resultado parece grande.	Escribir MDE, alpha, potencia y n antes de empezar.
Pasar de experimento a 100%	El equipo quiere cerrar rápido.	Usar rollout por pasos y rollback si falla un guardrail.
Confundir métrica primaria con métrica diagnóstica	Todas parecen interesantes.	Una primaria decide; las demás explican o bloquean.
Ignorar clústeres naturales	La unidad individual da más muestra aparente.	Preguntar si equipo, empresa, aula, operador o cola comparten efectos.
Ajustar observacional sin mirar solapamiento	El modelo devuelve un número.	Revisar si existen casos comparables antes de hablar de efecto.

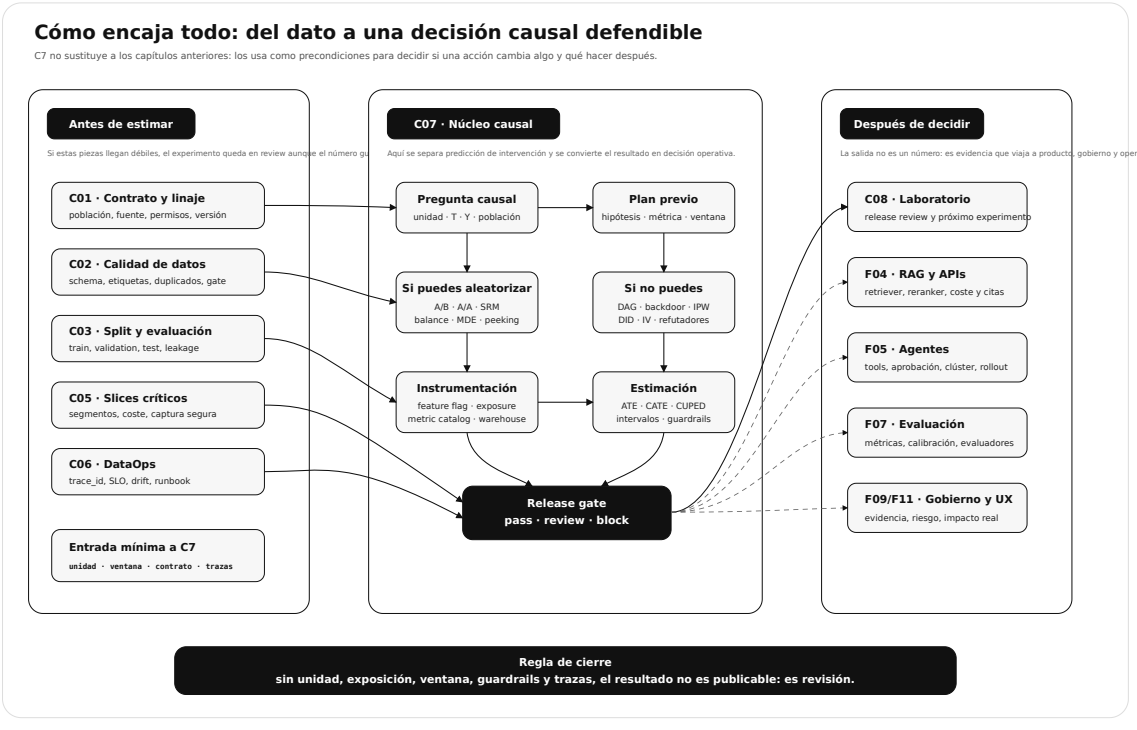
Cómo encaja todo

Este mapa se lee desde los capítulos anteriores. Los datos y contratos vienen del capítulo 01, la evaluación honesta del capítulo 03, los slices del capítulo 05 y la operación del capítulo 06. Este capítulo enseña a decidir si una acción cambia algo.

La decisión que introduce es nueva: no basta con medir calidad predictiva; necesitamos estimar efecto, revisar diseño experimental y dejar una salida operativa. En el cierre del facsímil, esto se convertirá en cierre integrador.



El Mermaid anterior sirve para estudiar relaciones. El SVG siguiente lo baja a arquitectura de decisión: qué piezas deben llegar sanas desde datos, qué decide este capítulo y qué salidas deja para laboratorio, gobernanza y producto. Es el dibujo que conviene tener en la cabeza cuando alguien dice “la variante ganó” demasiado rápido.



C7 encaja como la capa de decisión causal: recibe datos auditables, define intervención y exposición, estima efecto con incertidumbre y entrega una decisión que el laboratorio, la gobernanza y el producto pueden defender.

Vocabulario aprendido

TÉRMINO	DEFINICIÓN BREVE
Tratamiento	Acción o variante cuyo efecto queremos medir.
Control	Condición de comparación sin la intervención nueva.
Resultado	Variable que medimos después de la intervención.
Unidad	Elemento que se asigna y analiza: usuario, ticket, empresa o sesión.
ATE	Efecto medio del tratamiento.
CATE	Efecto medio condicionado a un contexto o segmento.
Contrafactual	Resultado que no observamos para la misma unidad bajo otra acción.
Confounder	Variable que afecta al tratamiento y al resultado.
DAG causal	Grafo que explicita hipótesis sobre relaciones causales.
SRM	Desajuste entre asignación esperada y observada.
Guardrail	Métrica que no debe empeorar aunque la primaria mejore.
CUPED	Ajuste con covariable previa para reducir varianza.
Solapamiento	Existencia de tratamiento y control dentro de contextos comparables.
Feature flag	Interruptor controlado en ejecución para servir variantes sin desplegar código nuevo.
Evaluation context	Contexto usado para evaluar una flag: unidad, segmento, canal, entorno o atributos.
Exposure event	Evento que demuestra que la unidad recibió la variante.
A/A test	Prueba con variantes equivalentes para validar instrumentación y reparto.
MDE	Efecto mínimo detectable bajo una muestra y potencia dadas.
Peeking	Mirar resultados repetidamente sin regla previa.
Rollout	Publicación progresiva con pasos, guardrails y rollback.
Plan de análisis	Contrato previo con hipótesis, población, métrica, ventanas, exclusiones y reglas de decisión.

TÉRMINO	DEFINICIÓN BREVE
Catálogo de métricas	Registro versionado que define nombre, unidad, ventana, dirección y propósito de cada métrica.
Comparaciones múltiples	Riesgo de encontrar señales aparentes al mirar muchas métricas, slices o ventanas.
Propensity score	Probabilidad estimada de recibir tratamiento dado el contexto observado.
Asignación por clúster	Diseño que asigna grupos completos cuando las unidades pueden influirse entre sí.
Efecto de diseño	Factor que aproxima cuánto aumenta la varianza cuando las observaciones dentro de un clúster se parecen entre sí.
Refutador causal	Prueba que intenta romper una conclusión causal con placebo, control negativo, subconjuntos o supuestos alternativos.
Control negativo	Variable o prueba que no debería cambiar por la intervención y sirve para detectar sesgo o mala instrumentación.
IPW	Ponderación por probabilidad inversa para ajustar por la probabilidad estimada de recibir tratamiento.
Estimador doblemente robusto	Estimador que combina modelo de resultado y propensity score para reducir dependencia de un único modelo.
Difference-in-differences	Diseño cuasi-experimental que compara cambios antes/después entre grupo tratado y grupo control.
Regression discontinuity	Diseño que estima efecto cerca de un umbral de asignación.
Variable instrumental	Variable que afecta al tratamiento y solo debería afectar al resultado a través de ese tratamiento.

Antes de pasar página

Antes de avanzar, deberías poder responder:

1. ¿Qué diferencia hay entre $P(Y \mid X)$ y $P(Y \mid do(X))$?
2. ¿Por qué no observamos $Y_i(1)$ y $Y_i(0)$ a la vez?
3. ¿Qué mide el ATE?
4. ¿Qué mide el CATE?
5. ¿Qué es un contrafactual?

6. ¿Qué piezas debe tener una pregunta causal?
7. ¿Qué comprueba SRM?
8. ¿Qué significa balance previo entre grupos?
9. ¿Por qué un guardrail puede bloquear un experimento ganador?
10. ¿Qué hace CUPED y qué no hace?
11. ¿Qué es un confounder?
12. ¿Por qué hace falta solapamiento en datos observacionales?
13. ¿Por qué un experimento puede quedar en `review` en lugar de `pass`?
14. ¿Por qué el efecto ingenuo observacional no basta?
15. ¿Qué demuestra un exposure event?
16. ¿Por qué un A/A test puede ahorrar un experimento mal medido?
17. ¿Qué significa MDE?
18. ¿Por qué mirar resultados muchas veces sin regla previa cambia el riesgo de error?
19. ¿Qué debería contener una política de rollout?
20. ¿Qué campos mínimos debe tener un plan de análisis?
21. ¿Qué diferencia hay entre métrica primaria, guardrail y diagnóstica?
22. ¿Cuándo usarías Bonferroni o control de descubrimientos falsos?
23. ¿Qué intenta resumir un propensity score?
24. ¿Por qué una asignación por clúster puede ser más honesta aunque tenga menos potencia?
25. ¿Qué significa efecto de diseño y por qué reduce tamaño efectivo?
26. ¿Qué diferencia hay entre confounder, mediador y collider?
27. ¿Qué intenta detectar un tratamiento placebo?
28. ¿Cuándo usarías IPW y cuándo desconfiarías de sus pesos?
29. ¿Por qué un estimador doblemente robusto no elimina la necesidad de buenos supuestos?
30. ¿Qué supuesto central necesita difference-in-differences?
31. ¿Qué condición fuerte necesita una variable instrumental?
32. ¿Cómo conecta este capítulo con el cierre del facsímil?

En resumen

IDEA	QUÉ TE LLEVAS
Predicción no es intervención.	Un score útil para anticipar no prueba que una acción cambie el resultado.
Un experimento es arquitectura.	Unidad, asignación, logging, métricas y análisis deben estar contratados.
El efecto necesita incertidumbre.	Un ATE observado sin intervalo puede llevar a decisiones precipitadas.
Los guardrails importan.	Una mejora primaria no justifica degradar coste, latencia o experiencia.
Causalidad observacional exige supuestos.	Si falta solapamiento o hay confounding, la lectura queda en triage.
Las decisiones se versionan.	El resultado útil es un artefacto: reporte, scorecard y decisión.
La plataforma importa.	Flags, exposición, métricas, warehouse y análisis forman parte del experimento.
El MDE evita autoengaños.	Una muestra pequeña puede aprender mucho y decidir poco.
El rollout también es causalidad aplicada.	Publicar progresivamente protege calidad, coste y operación.
El plan evita cambiar la pregunta.	Hipótesis, métrica, ventanas y reglas deben existir antes del resultado.
Los clústeres cambian el diseño.	Si hay aprendizaje compartido, la unidad individual puede engañar.
Las métricas necesitan contrato.	Nombre, unidad, ventana y dirección deben ser inequívocos.
El tamaño efectivo no siempre coincide con el número de filas.	Si hay dependencia dentro de equipos, colas o aulas, el efecto de diseño reduce la confianza real.
Los DAGs evitan ajustes peligrosos.	Confounder, mediador y collider piden decisiones distintas sobre columnas y filtros.
Los datos observacionales necesitan refutadores.	Placebo, control negativo, subsets y solapamiento ayudan a no creer demasiado pronto.
Los métodos avanzados no sustituyen supuestos.	IPW, AIPW, DID, regression discontinuity e instrumentos solo valen bajo condiciones defendibles.

Para saber más

Angrist, J. D., Imbens, G. W. y Rubin, D. B. (1996). Identification of Causal Effects Using Instrumental Variables. *Journal of the American Statistical Association*, 91(434), 444-455. DOI

Bang, H. y Robins, J. M. (2005). Doubly Robust Estimation in Missing Data and Causal Inference Models. *Biometrics*, 61(4), 962-973. DOI

Benjamini, Y. y Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B*, 57(1), 289-300. DOI

Austin, P. C. (2009). Balance Diagnostics for Comparing the Distribution of Baseline Covariates Between Treatment Groups in Propensity-Score Matched Samples. *Statistics in Medicine*, 28(25), 3083-3107. DOI

Casella, G. y Berger, R. L. (2002). *Statistical Inference* (2.ª ed.). Duxbury.

CausalML. (2026). CausalML Documentation. Documentación

Card, D. y Krueger, A. B. (1994). Minimum Wages and Employment: A Case Study of the Fast-Food Industry in New Jersey and Pennsylvania. *The American Economic Review*, 84(4), 772-793. JSTOR

Chernozhukov, V. y otros (2018). Double/Debiased Machine Learning for Treatment and Structural Parameters. *The Econometrics Journal*, 21(1), C1-C68. DOI

Chow, S.-C., Shao, J. y Wang, H. (2008). *Sample Size Calculations in Clinical Research* (2.ª ed.). Chapman and Hall/CRC.

Deng, A., Xu, Y., Kohavi, R. y Walker, T. (2013). Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data. *WSDM*, 123-132. PDF

Donner, A. y Klar, N. (2000). *Design and Analysis of Cluster Randomization Trials in Health Research*. Arnold.

Eppo. (2026). The Eppo Docs. Documentación

Evidently AI. (2026). Evidently Documentation. Documentación

GrowthBook. (2026). GrowthBook Documentation. Documentación

Gupta, S., Ulanova, L., Bhardwaj, S., Dmitriev, P., Raff, P. y Fabijan, A. (2018). The Anatomy of a Large-Scale Experimentation Platform. *IEEE International Conference on Software Architecture*. Microsoft Research

Holland, P. W. (1986). Statistics and Causal Inference. *Journal of the American Statistical Association*, 81(396), 945-960. DOI

Horvitz, D. G. y Thompson, D. J. (1952). A Generalization of Sampling Without Replacement From a Finite Universe. *Journal of the American Statistical Association*, 47(260), 663-685. DOI

Imbens, G. W. y Rubin, D. B. (2015). *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press. DOI

Johari, R., Pekelis, L. y Walsh, D. J. (2017). Peeking at A/B Tests: Why It Matters, and What to Do About It. *KDD*, 1517-1525. DOI

Kish, L. (1965). *Survey Sampling*. Wiley.

Kohavi, R., Longbotham, R., Sommerfield, D. y Henne, R. M. (2009). Controlled Experiments on the Web: Survey and Practical Guide. *Data Mining and Knowledge Discovery*, 18(1), 140-181. Microsoft Research

LaunchDarkly. (2026). Experimentation. Documentación

Lee, D. S. y Lemieux, T. (2010). Regression Discontinuity Designs in Economics. *Journal of Economic Literature*, 48(2), 281-355. DOI

Nie, K., Zhang, Z., Xu, B. y Yuan, T. (2022). Ensure A/B Test Quality at Scale with Automated Randomization Validation and Sample Ratio Mismatch Detection. *CIKM*. arXiv

O'Brien, P. C. y Fleming, T. R. (1979). A Multiple Testing Procedure for Clinical Trials. *Biometrics*, 35(3), 549-556. DOI

OpenFeature. (2026). Evaluation Context. Especificación

OpenFeature. (2026). Introduction. Documentación

Optimizely. (2026). Introduction to Optimizely Feature Experimentation. Documentación

Pearl, J. (2009). *Causality: Models, Reasoning, and Inference* (2.ª ed.). Cambridge University Press.

PostHog. (2026). Creating an Experiment. Documentación

Pocock, S. J. (1977). Group Sequential Methods in the Design and Analysis of Clinical Trials. *Biometrika*, 64(2), 191-199. DOI

PyMC. (2026). Bayesian Non-parametric Causal Inference. Documentación

PyWhy. (2026). DoWhy documentation. Documentación

PyWhy. (2026). EconML documentation. Documentación

Rosenbaum, P. R. y Rubin, D. B. (1983). The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*, 70(1), 41-55. DOI

Rubin, D. B. (1974). Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies. *Journal of Educational Psychology*, 66(5), 688-701. DOI

Sharma, A. y Kiciman, E. (2020). DoWhy: An End-to-End Library for Causal Inference. arXiv

Statsig. (2026). Experiment Options. Documentación

Notas

1. Rubin, D. B. (1974). Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies. *Journal of Educational Psychology*, 66(5), 688-701. https://doi.org/10.1037/h0037350 ↩

2. Holland, P. W. (1986). Statistics and Causal Inference. *Journal of the American Statistical Association*, 81(396), 945-960. https://doi.org/10.1080/01621459.1986.10478354 ↩

3. Pearl, J. (2009). *Causality: Models, Reasoning, and Inference* (2.ª ed.). Cambridge University Press. ↩

4. Kohavi, R., Longbotham, R., Sommerfield, D. y Henne, R. M. (2009). Controlled Experiments on the Web: Survey and Practical Guide. *Data Mining and Knowledge Discovery*, 18(1), 140-181. https://doi.org/10.1007/s10618-008-0114-1 ↩

5. Gupta, S., Ulanova, L., Bhardwaj, S., Dmitriev, P., Raff, P. y Fabijan, A. (2018). The Anatomy of a Large-Scale Experimentation Platform. *IEEE International Conference on Software Architecture*. https://www.microsoft.com/en-us/research/publication/the-anatomy-of-a-large-scale-experimentation-platform/ ↩

6. OpenFeature. (2026). *Introduction*. https://openfeature.dev/docs/reference/intro/. Consultado el 7 de junio de 2026. ↩

7. OpenFeature. (2026). *Evaluation Context*. https://openfeature.dev/specification/sections/evaluation-context/. Consultado el 7 de junio de 2026. ↩

8. LaunchDarkly. (2026). *Experimentation*. https://launchdarkly.com/docs/home/experimentation. Consultado el 7 de junio de 2026. ↩

9. GrowthBook. (2026). *GrowthBook Documentation*. https://docs.growthbook.io/. Consultado el 7 de junio de 2026. ↩

10. Statsig. (2026). *Experiment Options*. https://docs.statsig.com/statsig-warehouse-native/features/experiment-options. Consultado el 7 de junio de 2026. ↩

11. Eppo. (2026). *The Eppo Docs*. https://docs.geteppo.com/. Consultado el 8 de junio de 2026. ↩

.2. PostHog. (2026). *Creating an experiment*. https://posthog.com/docs/experiments/creating-an-experiment. Consultado el 22 de junio de 2026. ↩

.3. Optimizely. (2026). *Introduction to Optimizely Feature Experimentation*. https://docs.developers.optimizely.com/feature-experimentation/docs/introduction. Consultado el 8 de junio de 2026. ↩

.4. Evidently AI. (2026). *Evidently Documentation*. https://docs.evidentlyai.com/introduction. Consultado el 22 de junio de 2026. ↩

.5. Donner, A. y Klar, N. (2000). *Design and Analysis of Cluster Randomization Trials in Health Research*. Arnold. ↩

.6. Kish, L. (1965). *Survey Sampling*. Wiley. ↩

.7. Casella, G. y Berger, R. L. (2002). *Statistical Inference* (2.ª ed.). Duxbury. ↩

.8. Nie, K., Zhang, Z., Xu, B. y Yuan, T. (2022). Ensure A/B Test Quality at Scale with Automated Randomization Validation and Sample Ratio Mismatch Detection. *CIKM*. https://doi.org/10.1145/3511808.3557087 ↩

.9. Austin, P. C. (2009). Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine*, 28(25), 3083-3107. https://doi.org/10.1002/sim.3697 ↩

0. Chow, S.-C., Shao, J. y Wang, H. (2008). *Sample Size Calculations in Clinical Research* (2.ª ed.). Chapman and Hall/CRC. ↩

1. Statsig. (2026). *Experiment Options*. https://docs.statsig.com/statsig-warehouse-native/features/experiment-options. Consultado el 7 de junio de 2026. ↩

2. Johari, R., Pekelis, L. y Walsh, D. J. (2017). Peeking at A/B Tests: Why It Matters, and What to Do About It. *KDD*, 1517-1525. https://doi.org/10.1145/3097983.3097992 ↩

3. Pocock, S. J. (1977). Group Sequential Methods in the Design and Analysis of Clinical Trials. *Biometrika*, 64(2), 191-199. https://doi.org/10.1093/biomet/64.2.191 ↩

4. O'Brien, P. C. y Fleming, T. R. (1979). A Multiple Testing Procedure for Clinical Trials. *Biometrics*, 35(3), 549-556. https://doi.org/10.2307/2530245 ↩

5. Benjamini, Y. y Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B*, 57(1), 289-300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x ↩

6. Deng, A., Xu, Y., Kohavi, R. y Walker, T. (2013). Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data. *WSDM*, 123-132. https://doi.org/10.1145/2433396.2433413 ↩

27. Microsoft Research. (2022). Deep Dive Into Variance Reduction. https://www.microsoft.com/en-us/research/articles/deep-dive-into-variance-reduction/ ↩

28. Sharma, A. y Kiciman, E. (2020). DoWhy: An End-to-End Library for Causal Inference. *arXiv:2011.04216*. https://arxiv.org/abs/2011.04216 ↩

29. PyWhy. (2026). *DoWhy documentation*. https://www.pywhy.org/dowhy/main/index.html. Consultado el 7 de junio de 2026. ↩

30. PyWhy. (2026). *EconML documentation*. https://www.pywhy.org/EconML/spec/overview.html. Consultado el 7 de junio de 2026. ↩

31. Rosenbaum, P. R. y Rubin, D. B. (1983). The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*, 70(1), 41-55. https://doi.org/10.1093/biomet/70.1.41 ↩

32. Imbens, G. W. y Rubin, D. B. (2015). *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press. https://doi.org/10.1017/CBO9781139025751 ↩

33. Horvitz, D. G. y Thompson, D. J. (1952). A Generalization of Sampling Without Replacement From a Finite Universe. *Journal of the American Statistical Association*, 47(260), 663-685. https://doi.org/10.1080/01621459.1952.10483446 ↩

34. Bang, H. y Robins, J. M. (2005). Doubly Robust Estimation in Missing Data and Causal Inference Models. *Biometrics*, 61(4), 962-973. https://doi.org/10.1111/j.1541-0420.2005.00377.x ↩

35. Card, D. y Krueger, A. B. (1994). Minimum Wages and Employment: A Case Study of the Fast-Food Industry in New Jersey and Pennsylvania. *The American Economic Review*, 84(4), 772-793. https://www.jstor.org/stable/2118030 ↩

36. Lee, D. S. y Lemieux, T. (2010). Regression Discontinuity Designs in Economics. *Journal of Economic Literature*, 48(2), 281-355. https://doi.org/10.1257/jel.48.2.281 ↩

37. Angrist, J. D., Imbens, G. W. y Rubin, D. B. (1996). Identification of Causal Effects Using Instrumental Variables. *Journal of the American Statistical Association*, 91(434), 444-455. https://doi.org/10.1080/01621459.1996.10476902 ↩

38. PyWhy. (2026). *DoWhy documentation*. https://www.pywhy.org/dowhy/main/index.html. Consultado el 7 de junio de 2026. ↩

39. Sutton, R. S. y Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2.ª ed.). MIT Press. https://incompleteideas.net/book/the-book-2nd.html ↩

40. CausalML. (2026). *CausalML Documentation*. https://causalml.readthedocs.io/. Consultado el 22 de junio de 2026. ↩

1. PyMC. (2026). *Bayesian Non-parametric Causal Inference*. https://www.pymc.io/projects/examples/en/latest/causal_inference/bayesian_nonparametric_causal.html. Consultado el 22 de junio de 2026. ↩