▲ **Airflows** · Credence

MANUAL DE USUARIO · DEFINIR PACKS DE DECISIÓN

La decisión es un cálculo. Este manual explica el cálculo.

Para la persona de negocio que define, afina y opera los packs de decisión: qué significa cada concepto —de la señal a la recompensa—, cómo se calcula la propuesta, cómo se firma y cómo aprende el sistema con los resultados.

> **La regla de oro:** misma evidencia → misma decisión, hoy y dentro de un año. Cada paso queda explicado y congelado con su evidencia; cada decisión se puede re-ejecutar bit a bit; y ninguna decisión surte efecto sin la firma de una persona.

Airflows Credence · instancia **https://credence.airflows.com** · contacto **phernandez@airflows.com**
Noviembre de 2026 · v1.0 · documento de uso interno

Índice

1 La idea en una página

2 El recorrido de una decisión (las siete etapas)

3 Los conceptos, uno a uno — señales · factores · reglas · acciones · escalaciones · recompensas · contexto · perfil de cálculo · el registro

4 Cómo se calcula la propuesta (la aritmética, paso a paso)

5 Aprobar, sustituir o rechazar: el veto humano

6 Ejemplo completo: el pack «fraude en pagos»

7 Aprendizaje: cómo se afina el pack con resultados

8 Errores comunes y trucos del oficio

9 Referencia rápida de atributos

1 La idea en una página

Credence es una capa de decisión **determinista y auditable**. No es un modelo de IA que «opina»: es **aritmética sobre el vocabulario de su negocio**. Usted describe ese vocabulario —qué señales hay, qué se cree sobre el expediente, qué se puede hacer y qué cuesta cada cosa— y el motor calcula, para **cada alternativa posible**, su **valor esperado en euros**. La alternativa con mayor valor se propone. Nunca se ejecuta sola: una persona la firma.

> **Lo que Credence NO es**
>
> No es un LLM decidiendo (el lenguaje queda para redactar y explicar), no es una caja negra estadística (cada número es una fila visible y editable), y no es un motor de reglas rígido (valora todas las alternativas y aprende de los resultados). Es la intersección: cálculo determinista con probabilidades declaradas y costes reales.

Tres propiedades que conviene tener presentes durante todo el manual:

- **Determinismo.** Misma evidencia + misma configuración → misma propuesta. Cada decisión congela su configuración (un *hash*), su semilla y la versión del motor.
- **Transparencia.** La propuesta llega con las creencias usadas, las reglas que dispararon y el valor de **todas** las vías — se ve lo que se descartó y por cuánto.
- **Mando humano.** Toda decisión nace `PROPOSED`. No existe el modo automático.

2 El recorrido de una decisión (las siete etapas)

Etapa	Nombre	Qué ocurre
1	**Señales**	El sistema que llama (o el agente) aporta la evidencia del expediente: cada dato, con la razón de su valor.
2	**Creencias**	El motor combina evidencia + reglas y produce probabilidades con origen declarado. Nunca certeza.
3	**Acciones**	Lo que se puede hacer (SMS, llamada, fraccionar, bloquear…), cada una con su coste real en euros.
4	**Valor**	Para cada alternativa (y combinación), el motor calcula su valor esperado neto: recompensas − costes.
5	**Registro**	La decisión se congela con toda su evidencia: creencias, reglas disparadas, tabla de alternativas, semilla y hash.
6	**Firma**	Una persona aprueba, sustituye o rechaza — con identidad y motivo. Aquí no hay atajo.
7	**Outcome**	El resultado real (pagó, asistió, fue fraude…) vuelve al sistema y alimenta la recalibración, bajo veto humano.

Las etapas 1–4 son el cálculo (este manual las desarrolla); las etapas 5–7 son el gobierno y el aprendizaje.

3 Los conceptos, uno a uno

Un **pack de decisión** es el vocabulario completo de una decisión de negocio repetida (recobrar un impago, priorizar una alerta, aprobar un crédito…). Se edita desde la herramienta administrativa — cada concepto es un conjunto de **filas** — y cambia surte efecto en la decisión siguiente, sin desplegar nada.

3.1 Señales (la evidencia)

Las señales son los **hechos del expediente**, aportados por quien llama en el momento de decidir. Cada señal es una **proposición con un valor de probabilidad entre 0 y 1** — no un dato en bruto.

- `pago_reciente = 0.9` → «estoy bastante seguro de que este cliente pagó hace poco».
- `patron_transaccional_raro = 0.4` → «el patrón es sospechoso con confianza media».

Cómo se definen en el pack: solo el **nombre** de la señal (p. ej. `incidentes_previos`). El valor lo trae cada llamada. Regla práctica: si un dato del expediente puede expresarse como «estoy X % seguro de que…», es una señal.

> **Importante**
> Una señal que ninguna regla usa es **decorativa**: moverla no cambiará nada. Toda señal debe alimentar al menos una regla (sección 3.3) para entrar en el cálculo.

3.2 Factores (las creencias ocultas)

Los factores son las **incertidumbres que la decisión quiere resolver**: ¿pagará? ¿es fraude? ¿la alerta es real? Cada factor es binario (ocurre o no) y se describe con dos números:

Atributo	Significado	Ejemplo y lectura
`prior`	Probabilidad **antes de mirar nada** (0–1). El punto de partida de la creencia.	`Pagara: 0.18` → «sin más datos, el 18 % de estos expedientes paga solo».
`pObs`	Precisión de observación (0.5–1). Cuánto mejora el conocimiento del factor al observarlo.	`0.85` → las señales relacionadas aclaran mucho; `0.5` → no aclaran nada.
`dynamic`	Si una acción puede **resolverlo** (sí/no). «Pagará» es dinámico (una oferta lo resuelve); «es VIP» no.	Solo los dinámicos pueden aparecer en `resolves` de acciones.

Un factor especial: el **absorbente** (`absorbing` del pack) cierra el episodio cuando se resuelve — «pagó», «caso cerrado». Truco avanzado: si necesita que las recompensas se cuenten «planas» (una sola vez, sin

ponderar por lo que ya ocurriría solo), use un absorbente que nadie resuelve, como un `CasoCerrado` con prior 0.

3.3 Reglas (de la evidencia a la creencia)

Las reglas **suben la creencia** de una proposición cuando sus premisas están activas. Es el puente entre señales y factores:

```
premisas activas → conclusión más probable  (× peso)
```

```
regla «reincidencia_por_incidentes»:
  premisas: [incidentes_previos]
  conclusión: RiesgoReincidencia
  peso: 0.65
```

Lectura: «si hay incidentes previos, sube la creencia de reincidencia». Con `incidentes_previos = 1.0`, la creencia de `RiesgoReincidencia` pasa de su prior 0.30 a ≈ 0.66.

- **Peso** (0–1): fuerza de la regla. Varias reglas sobre la misma conclusión se combinan (noisy-OR): dos indicios débiles suman.
- **Combinador** de premisas: `product` (todas cuentan, multiplicándose — por defecto) o `min` (manda la más débil).
- La conclusión puede ser un **factor** (mueve el cálculo) o una **proposición derivada** (útil como condición de recompensas o escalaciones).

3.4 Acciones (lo que se puede hacer)

Cada acción es un **recurso con coste**. La propuesta final puede combinar varias (acción conjunta) o ser vacía — **WAIT**: no actuar también es una alternativa valorada.

Atributo	Significado
`cost`	Coste real de ejecutarla, en euros de la misma moneda que las recompensas.
`resolves` + `probability`	«Esta acción resuelve el factor F con probabilidad P». P.ej. `FRACCIONAMIENTO_3P resolves Pagara p=0.58`.
`info` + `infoPObs`	Acción de **información**: no resuelve, pero mejora la precisión de observación (pedir documentación, llamar para confirmar). El motor la elige cuando su **valor de información** supera su coste.
`dampenedBy` + `dampen`	La eficacia cae cuando otro factor está activo: «la llamada pierde 0.22 de probabilidad si hay reincidencia».
`boostWith` + `boost`	Al revés: la acción rinde más si la acompaña otra («la oferta funciona mejor junto a una llamada: +0.05»).

En una acción conjunta, si varias acciones resuelven el mismo factor, cuenta la de mayor probabilidad base (empate: orden de declaración).

3.5 Escalaciones (el deterioro)

Describen cómo **empeoran solas** las cosas si no se actúa: una proposición **aparece** con cierta probabilidad cuando se cumple una condición, salvo si cierta acción está interveniendo.

```
escalación:
  prop: CuentaComprometida        # lo que aparece
  probability: 0.2                # con qué probabilidad por paso
  when: [senal_compromiso]        # condición (proposiciones activas)
  unlessAnyAction: [BLOQUEAR, LLAMAR_CLIENTE]   # ...salvo si alguna de estas interviene
```

3.6 Recompensas (el valor de cada mundo)

Las recompensas son la parte **económica** del pack: qué gana o pierde el negocio en cada situación. Cada una es un componente declarativo que se **suma** al valor de una alternativa cuando se cumplen sus condiciones:

```
recompensa = cantidad × máscaras de estado × máscaras de acción × escalado × [× P(resolver)]
```

Bloque	Atributos	Lógica
Cantidad	`amount`	Euros (o multiplicador si va con escalado por contexto). Puede ser negativa (una pérdida).
Máscaras de estado ¿cuándo cuenta?	`when` / `whenAny` / `whenNot`	`when`: todos activos (AND, ponderado por la creencia de cada uno) · `whenAny`: alguno (OR) · `whenNot`: ninguno activo.
Máscaras de acción ¿con qué acción cuenta?	`withAnyAction` / `withAllAction` / `withoutAnyAction`	`withAny`: si alguna de esas acciones va en la alternativa · `withAll`: si van todas · `without`: solo si no va ninguna (así se premia/castiga el WAIT). Solo se usa una por recompensa.
Escalado	`scaleByContext` · `contextLinear` · `scaleByUncertaintyOf`	Multiplica por un escalar del contexto · por una suma ponderada de contextos · por la incertidumbre de un factor (el «valor de informarse»).
Dinámica	`timesResolveProbability` (+ `timesResolveOf`)	Multiplica por la probabilidad de que la acción resuelva el factor: convierte la recompensa en valor esperado de actuar.

```
recompensa «recupera_importe»:
  amount: 1.0
  withAnyAction: [LLAMADA_AGENTE]
  scaleByContext: saldo_eur          # × el saldo del expediente
  timesResolveProbability: true
  timesResolveOf: Pagara             # × P(la llamada consigue el pago)
```

Lectura: «si la alternativa incluye llamar, gana 1 × saldo × P(la llamada resuelve el pago)». Con saldo 2.400 € y P = 0.44 → **1.056 €** (menos el coste de la llamada, 4,20 → 1.052 € netos).

3.7 Contexto (los escalares del expediente)

El contexto lleva los **números del expediente** que escalan el valor: `saldo_eur`, `importe_eur`, `slot_value_eur`… No son creencias: son magnitudes ciertas. El puente `contextFromBelief` permite además derivar un contexto *desde* una creencia («annoy_p = P(cliente molesto)») para escalar recompensas con ella.

3.8 Perfil de cálculo (el motor)

Perfil	Qué hace	Cuándo usarlo
Compacto	Exploración completa y óptima (hasta ~512 combinaciones estado·acción).	Packs pequeños: el estándar.
Equilibrado	Simulación de escenarios acotada.	Packs medianos.
Intensivo	Búsqueda de alta capacidad (escala; replay sellado).	Packs grandes.
Directo	Cálculo inmediato sin búsqueda (y degradado de seguridad si otro perfil falla).	Latencia crítica.

Parámetros avanzados (los toque solo con acompañamiento): `depth` = cuántos pasos mira por delante · `discount` = cuánto vale el futuro respecto al presente (0–1) · `eta`, `episodes`, `budget` = esfuerzo de búsqueda. **Recomendación de negocio: depth 1–2 y no tocar el resto.**

3.9 El registro (la decisión congelada)

Cada decisión queda guardada con: **creencias usadas**, **reglas disparadas**, la **tabla completa de alternativas con su valor**, la **semilla**, el **config_hash** (huella de la configuración del pack) y la versión del motor. Eso permite el **replay**: re-ejecutar cualquier decisión pasada bit a bit. Si entre tanto el pack cambió (un coste, un umbral), el replay lo señala como **CONFIG DRIFT** — la desviación se detecta, no se descubre.

4 Cómo se calcula la propuesta (la aritmética, paso a paso)

Con todos los conceptos sobre la mesa, el cálculo completo de una decisión es este:

1. **Proyección.** Las creencias parten de los `prior` y suben con la evidencia y las reglas (noisy-OR).
2. **Valoración.** Para cada alternativa *a* (acción, combinación de acciones, o WAIT), el motor suma todas las recompensas cuya máscara de acción incluye *a*, ponderando las máscaras de estado por las creencias actuales, aplicando el escalado por contexto, y restando los costes de las acciones de *a*:

$$Q(a) = \Sigma \text{ recompensas}(a) - \Sigma \text{ costes}(a)$$

3. **Horizonte.** Con `depth` > 1 el motor además mira lo que pasa después (informarse hoy para decidir mejor mañana). Con depth 1, la valoración es la de un paso.
4. **Propuesta.** La alternativa con mayor Q se propone como `PROPOSED`, con toda la tabla visible.

Ejemplo guiado: el recobro del deck (2.400 €, 45 días, 2 incidentes, ingresos estables)

El pack tiene una recompensa por canal, con las probabilidades «versionadas» de la cartera, y penalizaciones que impiden mezclar canales (son alternativas):

```
Q(FRACCIONAMIENTO_3P) = 0.58 × 0.95 × 2.400 − 3,00  = 1.319 €   ← se propone
Q(LLAMADA_AGENTE)     = 0.44 × 1.00 × 2.400 − 4,20  = 1.052 €
Q(CARTA_CERTIFICADA)  = 0.41 × 1.00 × 2.400 − 8,50  =   976 €
Q(SMS_RECORDATORIO)   = 0.26 × 1.00 × 2.400 − 0,05  =   624 €
Q(WAIT)               = 0.18 × 1.00 × 2.400          =   432 €   (pago espontáneo esperado)
```

Fíjese en tres detalles que hacen al sistema honesto: el fraccionamiento recupera **2.280 €** si tiene éxito (0.95 × saldo — fraccionar cobra peaje), por eso gana con 1.319 y no con 1.389; **no actuar también tiene valor** (432 €: lo que se espera cobrar solo); y la **diferencia de 887 €** entre fraccionar y no actuar *es el valor de la decisión*. Nada de esto lo programó nadie: sale de las filas del pack.

> **Dónde verlo en vivo**
>
> En la herramienta administrativa, **Simulador de decisiones**: pack `recobro`, señales `pago_reciente 0 · incidentes_previos 0.4 · ingresos_estables 1.0`, contexto `saldo_eur 2400`. La tabla de alternativas reproduce la de arriba al euro, y las creencias (Pagara 18 %, CapacidadPago 78 %, RiesgoReincidencia 48 %) se recalculan al mover los controles.

5 Aprobar, sustituir o rechazar: el veto humano

Toda decisión nace **PROPOSED**. El ciclo de vida y quién manda en cada paso:

Estado	Cómo se llega	Significado
`PROPOSED`	El motor calcula y registra.	Propuesta con evidencia congelada. No ejecuta nada.
`APPROVED`	Una persona **aprueba** (consola, API o agente con permiso).	Se autoriza ejecutar la acción propuesta. Queda actor y momento.
`OVERRIDDEN`	Una persona **sustituye** la acción por otra, con motivo.	El veto humano corrige al motor — y el motivo alimenta el afinado.
`REJECTED`	Una persona la **rechaza**, con motivo.	La propuesta se descarta. Todo queda trazado.
`EXECUTED`	Se marca **ejecutada** tras surtir efecto.	El expediente espera su outcome.

Los cambios de estado son **append-only**: cada aprobación/rechazo añade una fila con **quién**, **cuándo** y **por qué**. Los saltos ilegales se rechazan. Ante cualquier reclamación o auditoría, la historia no se reescribe: se re-ejecuta.

Los cuatro verbos de un agente

Cualquier agente autorizado (incluido el asistente del vehículo) opera con cuatro herramientas: `decide()` · `explain()` · `verify(record)` (replay) · `approve(actor)`. Puede proponer y explicar todo lo que quiera; la firma exige identidad, y las acciones caras siempre pasan por una persona.

6 Ejemplo completo: el pack «fraude en pagos»

Definimos aquí un pack real de principio a fin, tal como se introduce en el asistente, y seguimos dos expedientes con números del sistema en producción.

6.1 El vocabulario

```
PACK fraude — «Bloquear, friccionar o pasar en tiempo real»

SEÑALES (las aporta el sistema de pagos):          FACTORES:
  dispositivo_nuevo                                 Fraude            prior 0.008  pObs 0.85
  geo_anomala                                       CuentaComprometida prior 0.010  pObs 0.80
(dinámico)
  patron_compra_raro                                Bloqueado         prior 0.001  pObs 0.95
(dinámico, absorbente*)
  tarjeta_fisica_presente
                                                   (*absorbente de cierre; el corte real es
CasoCerrado prior 0)

REGLAS:
  dispositivo_nuevo + geo_anomala → Fraude            peso 0.45
  patron_compra_raro              → Fraude            peso 0.35
  dispositivo_nuevo + geo_anomala → senal_compromiso  peso 0.50   (para la escalación)

ACCIONES:
  PASAR            coste 0
  FRICCION_3DS     coste 0.02   info  infoPObs 0.92
  LLAMAR_CLIENTE   coste 0.80   info  infoPObs 0.97
  BLOQUEAR         coste 4.00   resolves Bloqueado  probability 0.90

ESCALACIÓN:
  aparece CuentaComprometida (p 0.20) si senal_compromiso,
  salvo si interviene BLOQUEAR o LLAMAR_CLIENTE

RECOMPENSAS:
  evitar_cargo_fraudulento   +1.0 × importe   when [Fraude]  con BLOQUEAR
  fraude_pasado              −1.0 × importe   when [Fraude]  sin ninguna de las cuatro   (dejar
pasar el fraude)
  bloqueo_indebido           −0.15 × importe  whenNot [Fraude] con BLOQUEAR
(bloquear al cliente bueno)
  friccion_cliente           −0.40            whenNot [Fraude] con FRICCION_3DS o LLAMAR
(molestar al cliente bueno)
```

6.2 Expediente A: la transacción sospechosa (850 €)

Un pago con dispositivo nuevo (0.95) desde una geografía anómala (0.90) y patrón de compra raro (0.30), sin tarjeta física.

```
decide(fraude, expediente A, importe 850 €)
creencias →  Fraude 0.45 · CuentaComprometida 0.01 · Bloqueado 0.001

BLOQUEAR + FRICCION_3DS      Q =  596.19   ← propuesta
BLOQUEAR + LLAMAR_CLIENTE    Q =  595.41
BLOQUEAR + FRICCION + LLAMAR Q =  595.39

lectura: con Fraude al 45 %, el cargo fraudulento esperado (0.45×850 ≈ 383 €)
cae del lado de "no bloquear"; bloquear cuesta 0.15×850×0.55 ≈ 70 € si la tx
resulta legítima — sale a cuenta bloquear, y la fricción 3DS acompaña gratis.
```

6.3 Expediente B: la transacción legítima (850 €)

Mismo importe, cliente conocido: dispositivo habitual (0.05), geo normal (0.05), patrón normal (0.05).

```
decide(fraude, expediente B, importe 850 €)
creencias →  Fraude 0.026

FRICCION_3DS                 Q = −15.20   ← propuesta
LLAMAR_CLIENTE               Q = −15.97
(no actuar / PASAR)          Q ≈  −22.10

lectura: con Fraude al 2.6 %, bloquear ya no sale a cuenta (el castigo por
bloqueo indebido, 0.15×850×0.97 ≈ 124 €, supera el cargo evitable, 22 €).
La fricción 3DS, baratísima, se queda como red de seguridad. MISMO PACK,
otro expediente, OTRA decisión: el sistema se adapta al caso, no al umbral.
```

> **Lo que enseña este ejemplo**
>
> Un único pack, sin umbrales fijos, produce decisiones opuestas para expedientes opuestos — y en cada caso se puede señalar **qué fila** del pack produjo cada euro. Eso es lo que un comité de riesgos puede auditar.

7 Aprendizaje: cómo se afina el pack con resultados

El bucle de mejora tiene tres palancas, de la más automática a la más deliberada:

7.1 Outcomes: lo que de verdad pasó

Cuando una decisión ejecutada produce resultado (el deudor pagó, la tx era fraude, el paciente asistió), se registra el **outcome**: la observación de cada factor (0/1) y el resultado realizado. El motor recalcula entonces la **creencia posterior** — una actualización bayesiana exacta (o por partículas, SIR, en packs grandes) sobre los factores del expediente:

```
observe(decisión #812, {Pagara: 1}, realized="PAGO_TOTAL")
  P(Pagara)  0.18 → 0.18   (factors_snapshot congelado: referencia del pasado)
  posterior para el expediente: P(paga | ya pagó) = 1.0
  la estadística agregada de outcomes alimenta el RETUNEO del pack
```

7.2 Retuneo: editar las filas (bajo veto humano)

Las probabilidades y costes del pack son **filas versionadas**. El afinado es un cambio de datos, auditado, no un despliegue:

1. **Mire la evidencia:** la pantalla de uso agrega decisiones y outcomes por pack y día. ¿La llamada rinde menos que su coste? ¿El SMS recupera más que en la tabla?
2. **Ajuste la fila** en la herramienta administrativa (`PlannerAction`, `BeliefProp`…). El cambio aplica a las decisiones **futuras** desde la siguiente llamada.
3. **Versione:** guarde una versión del pack (`PackVersion`) con etiqueta y autor — el historial de política queda escrito.
4. **Observe el drift:** el replay de las decisiones pasadas marcará **CONFIG DRIFT** sobre las que se tomaron con los números anteriores. Esto es una característica: la supervisión puede preguntar «¿qué habría cambiado con la política actual?» y la respuesta es exacta.

> **Buena práctica**
> Afinez de a poco, con outcomes suficientes (un evento no es una tendencia); deje el motivo escrito en la versión; y si un cambio es grande, valídalo primero en el **Simulador** (no registra decisiones: es mesa de prueba).

7.3 El orden correcto cuando algo «no se mueve»

1. ¿La señal alimenta una regla cuya conclusión es un **factor usado** (recompensa, amortiguación)? Si no, es decorativa.

2. ¿La recompensa escala por un contexto con valor? (Un `saldo_eur` a 0 hace que toda acción sea pérdida y gane «no actuar».)
3. ¿El factor absorbente es el que cree? Un absorbente muy probable «cierra» el episodio y aplana el valor de actuar.

8 Errores comunes y trucos del oficio

Síntoma	Causa típica	Remedio
«El pack no compila» (estado del pack)	Nombres rotos: regla que cita una señal inexistente, reward con una acción mal escrita…	La pantalla de estado lista **todos** los errores; corríjalos y vuelva a validar. Un pack roto no afecta a los demás.
Muevo una señal y no cambia nada	Señal decorativa (sin regla hacia un factor usado).	Conéctela: regla → factor → reward/amortiguación (sección 7.3).
Siempre propone «no actuar»	Contexto a cero/vacío (todas las acciones son coste puro), o recompensas menores que los costes.	Revise el contexto del expediente y la economía del pack (¿amount × P supera el coste?).
Siempre propone lo mismo «toque lo que toque»	Riesgo fijo: ninguna señal mueve el factor que castiga/premia.	Haga que la señal concluya el factor de riesgo y que una recompensa (o amortiguación) lo use.
Propone «todas las acciones a la vez»	Recompensas que se apilan en las combinaciones.	Los canales alternativos llevan penalización por par (mire el pack recobro) o máscaras exclusivas.
Los valores «no cuadran» con mi cálculo a mano	Horizonte > 1 (cuenta continuidades) o absorbente muy activo (pondera por caso abierto).	Para reproducir una tabla de un paso: depth 1 y absorbente nunca-activo.

9 Referencia rápida de atributos

Dónde	Atributo	Rango	Una frase
Factor	`prior`	0 – 1	Creencia inicial sin mirar nada.
	`pObs`	0.5 – 1	Cuánto aclaran las señales del factor.
	`dynamic`	sí / no	Si una acción puede resolverlo.
Regla	`peso` · `combinador`	0 – 1 · product/min	Fuerza de la regla · cómo se combinan las premisas.

Acción	`cost`	€ ≥ 0	Coste real de ejecutarla.
	`resolves` + `probability`	0 – 1	P de resolver el factor dinámico.
	`info` + `infoPObs`	0.5 – 1	Acción de información: precisión que aporta.
	`dampenedBy` + `dampen`	0 – 1	Pierde eficacia si el factor está activo.
	`boostWith` + `boost`	0 – 1	Gana eficacia acompañada.
Escalación	`when` / `unlessAnyAction`	nombres	Aparece si…, salvo si interviene…
Recompensa	`when/whenAny/whenNot`	nombres	Estado en el que cuenta (AND/OR/salvo).
	`withAny/withAll/withoutAnyAction`	nombres	Acción con la que cuenta (una sola de las tres).
	`scaleByContext` · `contextLinear`	claves	Escalado por el contexto del expediente.
	`timesResolveProbability`	sí/no	Convierte en valor esperado de actuar.
Solver	`depth` · `discount`	1–4 · 0–1	Pasos por delante · valor del futuro.

Para llevarse a una reunión

Credence convierte una decisión de negocio en **una tabla en euros donde todas las alternativas están valoradas**, con sus números en filas visibles, su evidencia congelada, su replay exacto y su firma humana. Lo demás —señales, reglas, amortiguaciones, outcomes— son las piezas con las que usted escribe esa tabla.

Airflows Credence · https://credence.airflows.com · phernandez@airflows.com · Noviembre 2026 · v1.0