



KYNDA

Make it Kind, Make it Real

Propuesta técnica — Plataforma Kynda

Diagnóstico del sistema actual y plan de evolución

1. Resumen ejecutivo

Kynda tiene hoy un MVP en producción (Next.js + Supabase + Vercel) con un piloto real (Easyfairs) y un modelo de datos **notablemente bien diseñado**: multi-tenant, roles, catálogo con formularios dinámicos, KPIs definidos por fórmula, máquina de estados de evidencias, cadena/pulsera y consentimiento RGPD. Quien lo modeló pensó en el producto completo.

El problema no está en los datos: está en que **las tres promesas centrales del deck no están conectadas en la capa de aplicación**:

El deck promete	El producto hace hoy
Verificación por IA ("no proof, no impact")	Validación manual de Kynda Ops (David/Violeta aprueban una a una)
Impacto auditable y trazable	Los KPIs (16.845 kg, 1.203 personas) se teclean a mano en el panel
Reporting audit-grade (SHA-256, ledger, XBRL)	Dashboards + "Exportar a Excel". La capa de auditoría no existe

Lo llamativo es que el esquema **ya prevé** las tres cosas (`ai_review_run` , `action_kpi_def.formula` , `evidence_media` con GPS/timestamp/checksum). La conclusión del diagnóstico es, por tanto:

Evolucionar y refactorizar sobre la base actual, no reconstruir. Los motores que faltan (verificación, cálculo de impacto, auditoría) se construyen encima del modelo de datos existente, y en paralelo se repara la experiencia de producto (funnel de activación, editor de campañas, nomenclatura).

Un dato que cambia la conversación: con los precios actuales de los modelos de visión (sección 6.5), **verificar una evidencia con IA cuesta entre 0,4 y 2 céntimos**. El plan Enterprise completo (10.000 evidencias/año) supondría **~200 € anuales de coste de IA**. La verificación automática no es un problema de coste: es un problema de diseño de precisión y responsabilidad, y eso es exactamente lo que esta propuesta resuelve.

El plan se organiza en cuatro fases: **Fase 0** (auditoría técnica y cierre de estimación, 1–2 semanas), **Fase 1** (núcleo creíble: verificación + KPIs calculados + funnel, ~6–10 semanas), **Fase 2** (audit-grade + amplificación + pulsera, ~6–9 semanas) y **Fase 3** (Enterprise: SSO, HRIS, XBRL, metering, ~7–11 semanas).

2. Estado actual (as-is)

2.1 Stack y superficies

Componente	Tecnología	Evidencia
Frontend	Next.js (App Router), build en Vercel	Chunks <code>_next/static</code> , cabeceras de despliegue
Backend / BD	Supabase (Postgres + Auth + Storage)	Esquema entregado; acceso de cuenta facilitado
Multi-tenant	Subdominio por cliente (<code>easyfairs.bekynda.com</code>)	<code>brand.slug</code> + <code>webapp_url</code> en esquema
Panel empresa	Inicio, Campañas, Catálogo, Validación, Impacto e informes, Participantes, Comunicaciones	Capturas del admin

Componente	Tecnología	Evidencia
App empleado	Login, registro, recuperación, entrada por pulsera (/c/enter)	Rutas observadas
Piloto	Easyfairs: campaña "Banco de Alimentos Verano 2026", 3 jornadas, 28 inscritos, 30 evidencias validadas	Capturas del admin

2.2 Inventario del modelo de datos (31 tablas, por dominio)

Tenancy e identidad — `brand`, `brand_plan`, `app_user`, `role`, `brand_membership`, `portal_allowlist`, `invitation`. Evaluación: sólido. RBAC real con 6 roles (SUPERADMIN, OPS, BRAND_ADMIN, BRAND_MARKETING, BRAND_ESG, USER) y ámbito GLOBAL/BRAND. `brand_id` denormalizado en casi todas las tablas: la base correcta para Row Level Security por tenant. `signup_mode` PUBLIC/RESTRICTED ya contempla los dos modos de alta.

Catálogo y plantillas — `campaign_blueprint`, `blueprint_action`, `blueprint_action_field`, `field_option`, `action_kpi_def`. Evaluación: la joya del esquema. `blueprint_action_field` es un form-builder tipado (INT/FLOAT/ENUM/BOOL/TEXT/TIME, min/max, `suspicious_threshold`, `ui_hint`, dependencias entre campos). `action_kpi_def` define KPIs con fórmula, unidad, agregación (SUM/COUNT/AVG/MAX) y visibilidad. Todo lo necesario para calcular impacto de forma reproducible **ya está modelado**.

Campañas y ejecución — `campaign`, `action` (= jornada en la UI), `action_enrollment`, `action_execution`. Evaluación: funcional pero con deuda. `campaign` acumula ~50 columnas y varios jsonb (`objectives`, `evidence_config`, `social_config`, `enterprise_config`, `kpi_defs`) que **duplican** lo que ya viven en los blueprints. La máquina de estados de `action_execution` (STARTED → EVIDENCE_SUBMITTED → IN_VALIDATION → CLARIFICATION_REQUESTED → APPROVED/REJECTED/FRAUD_BLOCKED) está bien planteada.

Evidencia y validación — `evidence`, `evidence_media`, `validation`, `ai_review_run`. Evaluación: preparado para lo que falta. `evidence_media` captura `checksum`, `taken_at`, `gps_lat/lng`; `action` lleva `location_lat/lng` y ventana temporal; `evidence` soporta reintentos (`attempt_no`, `max_retries`). `validation.kind` distingue AUTO/MANUAL y `ai_review_run` (modelo, versión, resultado). `AUTHENTIC/FRAUD/UNCLEAR/CONTENT_VIOLATION`, `confianza`, `detalle`) está **definido pero sin uso**.

Amplificación — `generated_content` (red, formato, copy, hashtags, asset), `share_event` (con `proof_url`). Evaluación: modelo correcto para social **asistido** (la persona publica; la plataforma genera y registra), que es el enfoque viable frente a las restricciones de las APIs de Meta/X/LinkedIn.

Cadena / pulsera — `chain_token` (código físico/digital, estados UNUSED→ACTIVE→CONSUMED/REVOKED), `chain` (posición 1–6), `chain_link` (persona por eslabón, transferencias). Evaluación: el "impact chain" del deck está completo a nivel de datos, incluida la métrica de longitud de cadena.

Cumplimiento y operación — `esrs_disclosure`, `sdg_goal`, `user_consent` (tipo, versión, IP, user-agent, revocación), `notification` (EMAIL/PUSH/IN_APP), `audit_log`, `ong`. Evaluación: base RGPD seria (consentimiento versionado, `retention_days`, tombstoning de media). `audit_log` es una tabla mutable: sirve como registro operativo, **no** como evidencia audit-grade (ver sección 8). `ong` existe pero sin portal propio.

2.3 Flujos observados en el piloto

- **Validación:** 30 evidencias, todas aprobadas manualmente por David/Violeta Kynda, con reintentos reales (#2 en dos casos) y SLA informal de horas. La cola de "Pendientes de validar" y el detalle por evidencia (fotos + datos + quién validó + motivo) ya existen en la UI.
- **Impacto:** la pestaña Impacto presenta KPIs por jornada como **inputs editables** (kilos, lotes, personas beneficiarias, valor económico) que el admin rellena y guarda. El funnel de conversión (35 registrados → 28 inscritos → 27 evidencias → 27 aprobadas) se calcula bien.

- **Activación:** Home muestra "0% de 0 invitados" y Comunicaciones "0 invitaciones": el piloto se pobló por registro público, no por el sistema de invitaciones (`invitation` sin uso). La métrica queda rota y el funnel de adquisición no se mide.
- **Reporting:** chips ESRS (S1-6, S3-1, S3-4) y ODS por campaña, con "Framework validado por empresa". Export: Excel.
- **Higiene de datos:** usuarios de prueba (gmails personales, "Violeta Prueba", "Jorge Registro") conviven con empleados reales de `easyfairs.com` en producción.

2.4 Lo que no se ha podido verificar sin repo [A verificar]

#	Supuesto	Riesgo si es falso
A1	Existen políticas RLS por <code>brand_id</code> en todas las tablas	Crítico: fuga de datos entre clientes
A2	Storage de <code>evidence_media</code> es privado con URLs firmadas	Alto: fotos de personas accesibles públicamente
A3	La lógica de negocio vive en el servidor (route handlers / Edge Functions), no solo en el cliente	Alto: validaciones eludibles
A4	Retención (<code>retention_days</code>) y borrado se ejecutan realmente (jobs)	Alto: incumplimiento RGPD
A5	Índices adecuados en tablas de alto volumen (<code>evidence</code> , <code>action_execution</code> , <code>audit_log</code>)	Medio: degradación a escala
A6	Calidad/estructura del código front (componentes, tipado, tests)	Medio: coste de evolución

La Fase 0 convierte esta tabla en hechos.

3. Diagnóstico: causas raíz de los dos dolores

3.1 "La IA, la auditoría y el CSRD no están a la altura"

1. **Verificación manual encubierta.** El diferenciador n.º 1 del deck depende hoy de dos personas aprobando fotos. No escala (Enterprise = 10.000+ empleados), no es vendible como IA y consume al equipo fundador.
2. **Impacto tectleado.** Los titulares que la empresa enseñará a su auditor los introduce ella misma sin trazabilidad al dato origen. Un auditor CSRD preguntará "¿de dónde sale 16.845 kg?" y hoy la respuesta es "lo escribimos nosotros".
3. **Audit-grade inexistente.** No hay hash, sello de tiempo, inmutabilidad ni export regulatorio. `audit_log` es editable por definición (tabla normal).

3.2 "El producto/UX no se adecua a los objetivos"

1. **Funnel de activación roto:** invitaciones sin usar, métricas de adquisición sin sentido ("0% de 0"), sin recordatorios ni secuencia de onboarding.
2. **Editor de campaña monolítico:** un formulario de 10 secciones sobre una mega-tabla, con doble fuente de verdad (blueprint vs campaign) y validaciones difíciles de mantener.
3. **Nomenclatura inconsistente:** "jornada" (UI) = `action` (BD); `blueprint_action` (plantilla de tarea) convive con `action` (sesión concreta). Confunde a producto, a desarrollo y a cualquier tercero que se incorpore.
4. **Sin i18n:** todo en español con mercados declarados UE/US/LatAm.

4. Estrategia: evolución con refactors quirúrgicos

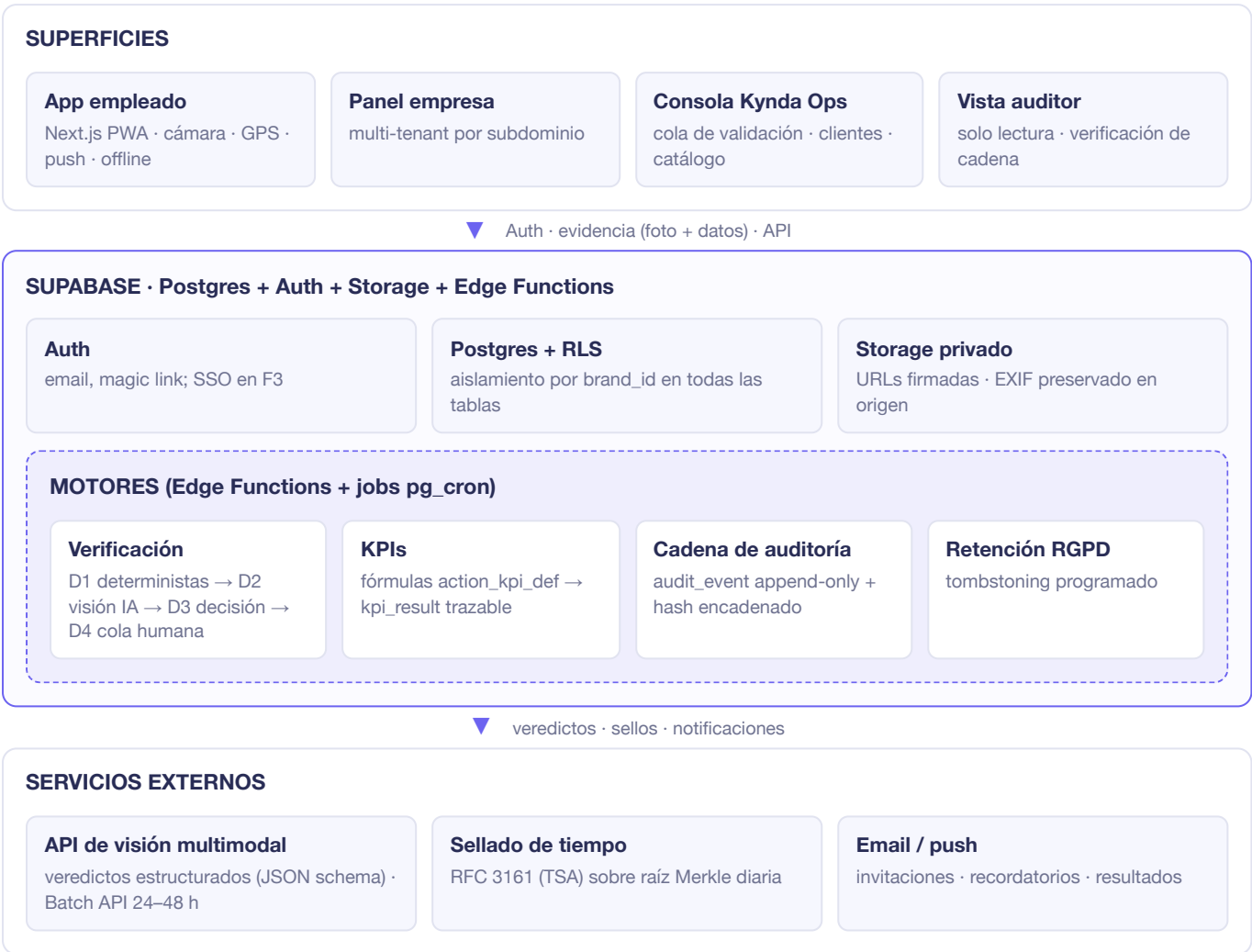
4.1 Veredicto

No reconstruir. El coste de tirar 31 tablas bien normalizadas y una app funcionando, para volver a llegar al mismo punto, no se justifica: los problemas están en la capa de aplicación. La reconstrucción total solo estaría justificada si la Fase 0 revelara ausencia total de RLS y un código de aplicación irrecuperable; incluso en ese escenario, el modelo de datos se conservaría.

4.2 Principios rectores

1. **Primero lo que hace creíble el producto:** verificación real y KPIs calculados. Es lo que separa a Kynda de un "portal de voluntariado bonito".
 2. **La IA nunca rechaza sola; solo aprueba sola.** Los rechazos siempre pasan por humano (responsabilidad y confianza; ver 6.4).
 3. **Una única fuente de verdad** por concepto: el blueprint define, la campaña hereda y sobrescribe explícitamente.
 4. **Trazabilidad de punta a punta:** cada número del dashboard debe poder explicarse hasta la evidencia (y su foto) que lo generó.
 5. **RGPD por diseño:** datos de empleados + GPS + imágenes de personas es la combinación más sensible posible en UE.
 6. **Sin sobre-ingeniería:** log hash-encadenado con sellado de tiempo en lugar de blockchain; social asistido en lugar de auto-publicación; PWA antes que apps nativas (decisión final en Fase 0).
-

5. Arquitectura objetivo



Decisiones clave por capa:

- **Front:** se mantiene Next.js/Vercel. La app de empleado se endurece como **PWA instalable** (Service Worker, cola offline de subida de evidencias con IndexedDB — el voluntariado ocurre en naves y calles con mala cobertura —, Web Push, acceso a cámara y geolocalización). Si el cliente exige apps nativas, la API queda lista y se decide en Fase 0.
- **Backend:** se mantiene Supabase. La lógica sensible (transiciones de estado, verificación, cálculo) se concentra en **Edge Functions y RPCs con security definer**, nunca en el cliente. `pg_cron` /colas para trabajos asíncronos.
- **RLS como línea de defensa principal:** políticas por tabla derivadas de `brand_membership` + rol (plantillas en 9.2), con suite de tests automatizada (pgTAP) que impide regresiones.

6. Motor de verificación (el diferenciador)

6.1 Pipeline por capas (de barato y determinista a caro y con juicio)

D0 — Ingesta y normalización. Al pasar `evidence` a SUBMITTED: extracción EXIF servidor (nunca confiar en el cliente), cálculo de `checksum` (SHA-256), registro del evento en la cadena de auditoría.

D1 — Checks deterministas (coste ~0; milisegundos):

Check	Regla	Código de motivo
Geocerca	<code>distancia(GPS foto, <code>action.location</code>) ≤ radio configurable</code>	<code>GEO_OUT_OF_RANGE</code>
Ventana temporal	<code>taken_at ∈ [<code>starts_at</code> - margen, <code>ends_at</code> + margen]</code>	<code>TIME_OUT_OF_WINDOW</code>
Metadatos	EXIF presente y coherente (dispositivo, timestamps internos)	<code>EXIF_MISSING</code> / <code>EXIF_INCONSISTENT</code>
Captura de pantalla	heurísticas de dimensiones/EXIF típicas de screenshot	<code>SCREENSHOT_SUSPECTED</code>
Duplicado exacto	<code>checksum</code> contra media de la persona, de la campaña y global	<code>DUPLICATE_MEDIA</code>
Campos	<code>input_data</code> contra min/max/ <code>suspicious_threshold</code> del blueprint	<code>FIELD_OUT_OF_RANGE</code>
Ráfaga	N envíos de la misma persona en ventana imposible	<code>BURST_ANOMALY</code>

D2 — Visión multimodal (solo si D1 no descarta): un modelo de visión evalúa con salida **estructurada y validada por esquema JSON** (sin parseo frágil): ¿la escena corresponde a la acción? ¿la actividad es plausible? ¿hay señales de manipulación, pantalla re-fotografiada, contenido inseguro o PII de terceros identificable? Resultado → `ai_review_run` (result, confidence, analysis_details, modelo y versión: reproducibilidad ante auditoría).

D3 — Matriz de decisión (configurable por campaña vía `control_level`):

D1	Resultado IA	Confianza	Decisión
PASS	AUTHENTIC	≥ 0,90	Auto-aprobación (<code>validation.kind = AUTO</code>)
PASS	AUTHENTIC	0,70–0,90	Cola humana, prioridad baja
PASS	UNCLEAR	—	Cola humana, prioridad media
WARN (1 check blando)	AUTHENTIC	≥ 0,90	Cola humana, prioridad baja
cualquiera	FRAUD	≥ 0,85	Cola humana, prioridad alta (nunca auto-rechazo)
cualquiera	CONTENT_VIOLATION	—	Bloqueo temporal + revisión humana
FAIL duro (duplicado exacto, GPS » radio)	—	—	<code>NEEDS_CLARIFICATION</code> al empleado (reintento)

D4 — Cola humana: la pantalla de Validación existente, alimentada solo con el ~10–25% dudoso, ordenada por `priority` y `sla_deadline`, mostrando los motivos de D1/D2 para que la persona revise en segundos, no minutos. Cada decisión humana registra reviewer, motivo y timestamp.

6.2 Flujo de reintento y apelación

Rechazo → notificación al empleado con motivo legible → puede subir nueva evidencia (`attempt_no` +1, máx. `max_retries`) → historial completo conservado. Con esto el "validaremos en 24–48 h" del deck pasa a ser: **mayoría en segundos (auto), resto < 24 h (cola priorizada)**.

6.3 Antifraude (más allá de una foto)

Dedup perceptual (pHash además del checksum exacto, detecta re-encodes), coherencia de dispositivo por usuario, patrones de campaña (misma foto desde varias cuentas), y señales agregadas por empresa para Kynda Ops.

6.4 Responsabilidad y gobernanza

- La IA **no rechaza ni acusa**: solo aprueba con alta confianza o deriva a humano. Un falso positivo que llegue a un informe CSRD auditado es un riesgo legal; por diseño, todo APPROVED es o (a) IA con confianza alta + checks OK, o (b) decisión humana registrada.
- Umbrales por campaña (`control_level` LOW/MEDIUM/HIGH ya existe en el esquema) y por cliente.
- Registro completo en la cadena de auditoría: qué modelo, qué versión, qué confianza, qué humano.

6.5 Modelo de costes de la IA (precios oficiales API Anthropic, julio 2026)

Supuesto por evidencia: 1 foto (~1.600 tokens a resolución estándar) + prompt y campos (~900 tokens) + veredicto estructurado (~300 tokens de salida).

Modelo	Precio (in/out por M tokens)	Coste por evidencia	10.000 evidencias/año
Haiku 4.5 (triaje)	\$1 / \$5	~\$0,004	~\$40
Sonnet 5	\$3 / \$15	~\$0,012	~\$120
Opus 4.8 (máxima precisión)	\$5 / \$25	~\$0,020	~\$200
Cualquiera vía Batch API (SLA 24 h, -50%)		mitad	mitad

Lecturas:

1. **El coste de IA es despreciable** frente al precio de los planes (35k–130k €/año). Incluso usando el modelo más capaz para todo, el plan Enterprise consume ~200 \$/año de inferencia. La validación manual actual es el único coste que no escala.
2. Recomendación: **arrancar con un único modelo de máxima precisión** para todo (simplicidad + mejor calidad donde hay responsabilidad legal), medir en el piloto, e introducir triaje barato (Haiku) solo si el volumen lo justificara. La cola no urgente puede ir por Batch API con el SLA de 24–48 h que ya comunica el deck.
3. Fotos adicionales suben el coste linealmente (2–3 fotos ≈ 2–3× input): sigue siendo céntimos.
4. La decisión final de proveedor/modelo se cierra en la **PoC de Fase 0** (ver 11.1) con las 30 evidencias reales del piloto: precisión medida, no prometida.

7. Motor de KPIs e impacto calculado

7.1 Principio

Ningún KPI de titular se teclea. Cada `action_kpi_def.formula` se evalúa sobre el `input_data` de evidencias **APPROVED** (parser seguro de expresiones con lista blanca de funciones; nunca `eval`), se agrega según `aggregation` y se materializa con trazabilidad:

```
CREATE TABLE kpi_result (  
  id uuid PRIMARY KEY DEFAULT gen_random_uuid(),  
  brand_id uuid NOT NULL REFERENCES brand(id),  
  campaign_id uuid NOT NULL REFERENCES campaign(id),  
  action_id uuid REFERENCES action(id),          -- NULL = agregado de campaña  
  kpi_key text NOT NULL,  
  value numeric NOT NULL,  
  unit text NOT NULL,  
  evidence_count integer NOT NULL,  
  formula_snapshot text NOT NULL,                -- fórmula usada (versionado)  
  inputs_hash text NOT NULL,                     -- sha256 del conjunto de entradas  
  computed_at timestamptz NOT NULL DEFAULT now()  
);  
  
CREATE TABLE kpi_result_evidence (              -- drill-down KPI → evidencias  
  kpi_result_id uuid REFERENCES kpi_result(id),  
  evidence_id uuid REFERENCES evidence(id),  
  PRIMARY KEY (kpi_result_id, evidence_id)  
);
```

Recomputación por evento (aprobación/revocación de evidencia) + job de reconciliación nocturno. `action.results` queda como caché de lectura.

7.2 Overrides auditados

Si la empresa necesita corregir un dato (p. ej. el banco de alimentos reporta el pesaje oficial), se registra un `kpi_override` con autor, motivo y valor anterior — y aparece en la cadena de auditoría y en el informe como "dato ajustado por la organización". Transparencia en lugar de edición silenciosa.

7.3 Metodologías declaradas

"Personas beneficiarias equivalente" y "valor económico del voluntariado" son **estimaciones por factor** (p. ej. €/hora según estándar elegido). La metodología se configura por KPI (factor, fuente, versión) y se imprime en los informes. Un auditor no penaliza una estimación; penaliza una estimación sin metodología. El contenido regulatorio (qué factor usar, mapeo acción→datapoint ESRS) lo valida Kynda como experta ESG; la plataforma lo hace configurable y trazable.

7.4 UI

La pestaña Impacto pasa de inputs editables a: valor calculado + nº de evidencias que lo componen + drill-down hasta cada evidencia (foto, datos, quién/qué lo validó) + indicador de overrides. Es la demo más potente frente a un CSO: "haz clic en el número y te llevo hasta la foto".

8. Capa audit-grade y reporting CSRD

8.1 Cadena de auditoría (sin blockchain)

Un log **append-only con hash encadenado** cubre "auditable, trazable, verificable" sin la complejidad operativa de una DLT:

```
CREATE TABLE audit_event (  
  seq bigint GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  brand_id uuid NOT NULL,  
  stream text NOT NULL,          -- 'evidence' | 'validation' | 'kpi' | 'report' | 'consent'  
  entity_type text NOT NULL,  
  entity_id uuid NOT NULL,  
  event_type text NOT NULL,      -- 'evidence.approved', 'kpi.computed', ...  
  payload jsonb NOT NULL,        -- canónico (claves ordenadas)  
  payload_hash text NOT NULL,    -- sha256(payload canónico)  
  prev_hash text NOT NULL,       -- hash del evento anterior del mismo brand  
  hash text NOT NULL,            -- sha256(prev_hash || payload_hash || seq || ts)  
  created_at timestamptz NOT NULL DEFAULT now()  
);  
-- Inmutabilidad: REVOKE UPDATE/DELETE a todos los roles + trigger que aborta cambios.
```

- **Anclaje externo:** raíz de Merkle diaria sellada vía **RFC 3161 (TSA)**; opcionalmente publicada. Esto responde a "¿y si el proveedor edita su propia BD?" sin blockchain.
- **Verificador:** endpoint + CLI que recomputa la cadena y valida sellos; utilizable por el auditor del cliente.
- Si a futuro el mercado exigiera DLT, la cadena de hashes se ancla a una red permissioned sin rehacer nada.

8.2 Vista de auditor y exports

- **Vista de solo lectura por empresa** (usuario auditor invitado): campañas, KPIs con drill-down, cadena verificable, consentimientos.
- **Exports Fase 2:** CSV/JSON estructurado por datapoint ESRS + PDF de informe de campaña/periodo con metodologías.
- **XBRL/ESEF:** Fase 3. Es un formato regulado y pesado; lo correcto es validar antes el contenido con exports simples que los equipos ESG ya consumen.
- `esrs_disclosure` y `sdg_goal` se mantienen como tablas de referencia versionadas (la taxonomía ESRS evoluciona).

9. Producto, UX y datos

9.1 Reparaciones de producto (Fase 1)

Área	Cambio
Funnel de activación	Invitaciones e2e (<code>invitation</code> + emails con plantilla ya diseñada), estados invitado→registrado→inscrito→activo, métricas reales en Home, recordatorios automáticos (jornada próxima, evidencia pendiente)

Área	Cambio
Editor de campaña	Wizard por pasos con herencia explícita del blueprint (badge "heredado/personalizado" por campo), validación por paso, autosave. La mega-tabla <code>campaign</code> se aligera moviendo configuración a estructuras propias
Nomenclatura	Glosario único (ver Anexo A) aplicado a UI, código y documentación. En BD se mantienen los nombres (coste de migración > beneficio) pero el mapeo queda documentado
Limpieza	Purga de usuarios de prueba en producción; entorno demo separado con datos sintéticos
i18n	<code>next-intl</code> con es/en desde Fase 1 (el coste marginal ahora es mínimo; hacerlo tarde es caro)
Accesibilidad	AA como criterio de aceptación en componentes nuevos

9.2 Plantilla RLS (a validar/implantar en Fase 0-1)

```

CREATE FUNCTION is_member_of(target_brand uuid) RETURNS boolean
LANGUAGE sql STABLE SECURITY DEFINER AS $$
  SELECT EXISTS (
    SELECT 1 FROM brand_membership bm
    WHERE bm.brand_id = target_brand
      AND bm.user_id = auth.uid()
      AND bm.status = 'ACTIVE');
$$;

CREATE FUNCTION has_brand_role(target_brand uuid, roles text[]) RETURNS boolean
LANGUAGE sql STABLE SECURITY DEFINER AS $$
  SELECT EXISTS (
    SELECT 1 FROM brand_membership bm
    JOIN role r ON r.id = bm.role_id
    WHERE bm.brand_id = target_brand
      AND bm.user_id = auth.uid()
      AND bm.status = 'ACTIVE'
      AND r.code = ANY(roles));
$$;

-- Ejemplo (evidence): el empleado ve lo suyo; los roles de empresa, lo de su marca
ALTER TABLE evidence ENABLE ROW LEVEL SECURITY;
CREATE POLICY evidence_select_own ON evidence FOR SELECT USING (
  EXISTS (SELECT 1 FROM action_execution ae
    WHERE ae.id = evidence.action_execution_id
      AND ae.user_id = auth.uid()));
CREATE POLICY evidence_select_brand ON evidence FOR SELECT USING (
  has_brand_role(brand_id, ARRAY['BRAND_ADMIN', 'BRAND_ESG', 'BRAND_MARKETING']));

```

Storage: bucket privado `evidence/{brand_id}/{campaign_id}/{evidence_id}/...` con política por prefijo + URLs firmadas de corta duración. EXIF se conserva en origen (lo necesita el motor) y se **elimina en cualquier derivado compatible**.

9.3 RGPD operativo

- Consentimientos en el flujo real (TERMS/PRIVACY al alta; IMAGE_USE por campaña antes de compartir), versionados (ya soportado por `user_consent`).
- Job de retención: cumplido `retention_days` → tombstoning de media (borra objeto, conserva hash para la cadena) — el esquema ya tiene `tombstoned_at`.
- Derechos ARSOPL: export de datos del empleado y borrado/anonimización manteniendo agregados.
- DPA con procesadores (Supabase, Vercel, proveedor IA, email) y residencia UE de los proyectos. La política de uso de datos del proveedor de IA (no entrenamiento con datos de cliente) se documenta en el DPIA.
- El asesoramiento jurídico formal (DPIA firmado) corre del lado de Kynda; la plataforma aporta los mecanismos y la documentación técnica.

10. Cambios de modelo de datos propuestos (resumen)

Cambio	Tipo	Fase
<code>kpi_result</code> + <code>kpi_result_evidence</code> + <code>kpi_override</code>	Nuevas tablas	F1
<code>audit_event</code> (append-only) + anclaje TSA	Nueva tabla + job	F2 (esqueleto en F1)
<code>usage_ledger</code> (consumo de impact/story records por brand para facturación por uso)	Nueva tabla	F3
Aligerar <code>campaign</code> (mover <code>kpi_defs</code> , <code>evidence_config</code> , <code>social_config</code> a estructuras propias con herencia de blueprint)	Refactor	F1–F2
Índices en <code>evidence(brand_id, status)</code> , <code>action_execution(action_id, user_id)</code> , <code>audit_event(brand_id, seq)</code> , etc.	Índices	F0–F1
pHash en <code>evidence_media</code> (dedup perceptual)	Columna + job	F1

Nota: el deck factura por "impact records" y "story records", pero **no existe hoy ninguna tabla de metering**. `usage_ledger` conecta el producto con su propio modelo de ingresos (detalle en Fase 3).

11. Plan de entregas

11.1 Fase 0 — Auditoría técnica y cierre (1–2 semanas)

Entregables:

- 1. **Informe de seguridad:** estado real de RLS y Storage (A1–A2), revisión de Auth, rotación de credenciales, 2FA, gestión de secretos.
- 2. **Informe de código:** arquitectura del front/back real, deuda, tests existentes (A3, A6).
- 3. **PoC de verificación con las 30 evidencias reales del piloto:** pipeline D1+D2 ejecutado sobre datos reales → tasa de auto-aprobación, falsos positivos/negativos, coste medido por evidencia, elección de modelo. *Es también una pieza de venta para vosotros: "la IA ya validó vuestro piloto".*
- 4. **Decisiones cerradas:** PWA vs nativo, proveedor IA, alcance exacto de Fase 1.
- 5. **Backlog priorizado y estimación firme** de Fase 1 (sustituye a los rangos de este documento).

Requisitos: repo (solo lectura), proyecto Supabase (rol Developer), 1–2 sesiones con quien construyó el MVP.

11.2 Fase 1 — Núcleo creíble (~6–10 semanas con 2 devs)

Workstream	Contenido	Jornadas	Criterio de aceptación
W0 Fundaciones	Entornos dev/staging/prod, CI/CD (lint, types, tests, migraciones), seeds, suite RLS (pgTAP)	6–9	Pipeline verde obligatorio; tests RLS cubren todas las tablas con <code>brand_id</code>
W1 Verificación	D0–D4 completos, reason codes, reintentos, antifraude básico, cola humana renovada	21–30	≥70% de evidencias válidas auto-aprobadas en staging con datos del piloto; 0 auto-rechazos; toda decisión con motivo
W2 KPIs	Parser de fórmulas, <code>kpi_result</code> + trazabilidad, recomputación, UI con drill-down, overrides	15–21	Ningún KPI editable a mano; cada titular navegable hasta evidencia
W3 Funnel y editor	Invitaciones e2e, métricas de funnel, recordatorios, wizard de campaña, nomenclatura, limpieza	12–18	Funnel completo medible con datos reales; creación de campaña desde catálogo < 10 min
W4 RGPD	Consentimientos en flujo, job de retención, políticas Storage, export/borrado de usuario	4–6	Retención demostrable; DPIA técnicamente documentado
Total		58–84	

11.3 Fase 2 — Audit-grade, amplificación y pulsera (~6–9 semanas)

Workstream	Contenido	Jornadas
Cadena de auditoría	<code>audit_event</code> + inmutabilidad + Merkle diario + TSA + verificador	8–12
Vista auditor + exports	Rol auditor, informe PDF, export CSV/JSON por datapoint ESRS	6–9

Workstream	Contenido	Jornadas
Amplificación asistida	Generación de contenido por red (<code>generated_content</code>), editor según permisos de campaña, compartir nativo (Web Share API), registro <code>share_event</code>	8–12
Pulsera / cadena	Generación de lotes de códigos, hoja imprimible QR, flujo <code>/c/enter</code> completo, visualización de cadenas y KPI de longitud	8–12
Total		30–45

11.4 Fase 3 — Enterprise (~7–11 semanas)

SSO (SAML/OIDC), aprovisionamiento SCIM/import HRIS, multi-marca (vista holding), XBRL/ESEF, `usage_ledger` + integración de facturación (Stripe), rendimiento y particionado, API pública + webhooks. Total orientativo: **42–65 jornadas**.

11.5 Supuestos y exclusiones

- Estimaciones en jornadas de desarrollo efectivas; el calendario asume 2 desarrolladores. Precio = jornadas × tarifa acordada; los rangos se cierran tras Fase 0.
- Excluido: fabricación/logística de pulseras (la plataforma genera códigos y material imprimible), auto-publicación en redes (asistida por diseño), apps nativas (salvo decisión en F0), asesoría jurídica formal y mapeo regulatorio de contenido (lo aporta Kynda), migraciones de datos de terceros.
- Diseño: se trabaja sobre el design system existente (periwinkle/lima/salmón/rosa/navy); si hay Figma, se integra como fuente.

12. Calidad, entornos y operación

- **Entornos:** proyectos Supabase separados dev/staging/prod; migraciones versionadas en repo (supabase CLI) + tipos generados; Vercel Preview por PR.
- **Testing:** unit (lógica de motores), **pgTAP para RLS** (cada tabla con `brand_id` tiene test de aislamiento), e2e Playwright de los 5 flujos críticos (invitar → registrarse → inscribirse → evidenciar → verificar), test de carga de ráfagas de evidencias.
- **Observabilidad:** Sentry (front + functions), logs estructurados, panel interno de costes de IA por tenant, alertas de cola de validación y de fallos de sellado.
- **Colaboración:** sprints de 2 semanas con demo, tablero compartido, changelog por release; documentación técnica y runbooks en el repo.

13. Riesgos y mitigaciones

Riesgo	Prob.	Impacto	Mitigación
RLS ausente o incompleta en producción	? (F0)	Crítico	Auditoría inmediata; si falta, es lo primero que se implanta con suite de tests
Precisión IA insuficiente en categorías concretas de acción	Media	Alto	PoC F0 con datos reales; umbrales por campaña; cola humana como red; mejora iterativa de prompts por tipo de acción
Responsabilidad por dato erróneo en informe auditado	Baja	Alto	IA nunca rechaza/acusa sola; trazabilidad total; overrides auditados; metodologías declaradas
Dependencia de un proveedor de IA	Media	Medio	Contrato de datos adecuado; capa de abstracción del veredicto (esquema propio); segundo proveedor evaluado en PoC
Alcance de F1 crece (pulsera/social presionan)	Media	Medio	Fases cerradas; cambios via backlog priorizado con coste visible
Datos de prueba/reales mezclados	Ya ocurre	Medio	Limpieza F1 + entorno demo separado
Claves comprometidas (compartidas por email)	Ya ocurre	Alto	Rotación inmediata, gestor de secretos, 2FA, accesos por rol mínimo

14. Qué necesito de Kynda para arrancar

1. Acceso **solo lectura** al repositorio (o export del código).
2. Invitación al **proyecto** Supabase con rol Developer (no la cuenta propietaria) + rotación de las credenciales ya compartidas.
3. 1–2 sesiones de traspaso con quien construyó el MVP.
4. Decisión sobre PWA vs nativo si ya tenéis postura (si no, la traigo yo argumentada en F0).
5. Contacto ESG interno para validar metodologías y mapeos (sección 7.3).

Anexo A — Glosario y mapeo de nomenclatura

Término producto (propuesto)	UI actual	BD	Definición
Programa	Catálogo/Blueprint	<code>campaign_blueprint</code>	Plantilla reutilizable curada por Kynda
Campaña	Campaña	<code>campaign</code>	Instancia de un programa en una empresa, con fechas y objetivos
Sesión	Jornada	<code>action</code>	Convocatoria concreta (fecha, lugar, plazas) dentro de una campaña
Tarea tipo	—	<code>blueprint_action</code>	Definición de actividad y su formulario en el programa
Inscripción	Inscrito	<code>action_enrollment</code>	Reserva de plaza de un empleado en una sesión
Participación	—	<code>action_execution</code>	Ejecución real: de inscrito a verificado
Evidencia	Evidencia	<code>evidence (+ evidence_media)</code>	Prueba aportada (fotos + datos de formulario)
Impact record	—	evidencia APPROVED (unidad de facturación)	Acción verificada convertida en dato ESG
Story record	—	<code>generated_content</code>	Contenido de comunicación generado desde una evidencia
Cadena	Impact chain	<code>chain / chain_link / chain_token</code>	Secuencia de personas activadas por una pulsera

Anexo B — Referencias normativas presentes en el producto

ESRS: S1-6 (características de plantilla), S3-1 (políticas sobre comunidades afectadas), S3-4 (acciones sobre impactos materiales en comunidades) y catálogo E1-G1 en `esrs_disclosure`. ODS: 1, 2, 3, 10, 12, 13, 17 observados en el piloto. El mapeo acción→datapoint es configurable por campaña y validado por el equipo ESG de Kynda ("Framework validado por empresa" ya existe como concepto en la UI).