Conectia.pro

MUESTRA

Diagnóstico y Blueprint — muestra del entregable

Producto: conectia.pro, la web pública de Conectia PRO (React 18 + TypeScript, estática, prerenderizada, tres idiomas) **Fecha:** 17 de octubre de 2026 · **Autor:** Marc Molas, CTO · **Lector:** el fundador o CTO responsable del producto **Estado de este documento:** MUESTRA. Enseñamos el entregable sobre nuestro propio producto porque es la única base de código que podemos abrir entera sin tocar datos de ningún cliente. Todas las incidencias son reales; cada una está resuelta o en curso, con su ticket al lado.

> Esto es lo que recibes al final de las dos semanas: un veredicto por módulo, un registro de incidencias ordenado por gravedad, la arquitectura objetivo, el plan de la siguiente fase y los ficheros de contexto para tus agentes de programación. El presupuesto de la siguiente fase va en un documento aparte.

---

1. Alcance y acceso

- **Qué hemos leído:** el repositorio (enrutado, i18n, SEO, pipeline del blog, analítica, formularios, build y despliegue), la configuración de la CI, los informes de Search Console y de Core Web Vitals, y el despliegue en producción. Acceso de solo lectura en todo momento.
- **Qué hemos ejecutado:** el lint, la suite de tests, el build completo de producción (prebuild → comprobación de tipos → bundle de cliente → bundle de servidor → sitemap → prerender) y los flujos críticos en un navegador.
- **Qué no hemos tocado:** secretos, credenciales de analítica y la cuenta de hosting más allá de leer su configuración. Nunca pedimos contraseñas por chat.
- **Cómo se construyó el producto:** un equipo sénior trabajando con agentes de programación (Claude Code) bajo convenciones escritas: un `CLAUDE.md`, cuatro skills de proyecto y un prebuild que comprueba imágenes y metadatos SEO antes de que nada salga. Importa para el veredicto: el código es consistente, y los problemas son los que una demo nunca enseña.

---

2. Mapa de módulos

#	Módulo	Qué hace	Dónde vive
M1	Enrutado e i18n	Una sola configuración gobierna 64 rutas × 3 idiomas, slugs localizados y cambio de idioma sin perder la página	`src/utils/routes.config.ts`, `src/locales/`
M2	SEO y prerender	Canonical y hreflang salen de la configuración de rutas; 680 páginas se renderizan a HTML estático en un solo proceso Node, sin navegador headless	`src/components/seo/`, `scripts/prerender.mjs`
M3	Pipeline del blog	186 artículos por idioma en Markdown; el prebuild genera el índice y las imágenes OG	`src/content/blog/`, `scripts/generate-blog-meta.mjs`
M4	Analítica	Eventos propios vía GTM a GA4: secciones vistas, clics muertos, campos de formulario tocados, clics a WhatsApp. Sin herramienta de replay, sin banner de cookies	`src/utils/analytics.ts`, `docs/growth/tracking-plan.md`

#	Módulo	Qué hace	Dónde vive
M5	Contacto	Formulario nativo que envía a HubSpot más un enlace a WhatsApp; el calendario es el segundo paso, no el primero	`src/components/contact-…compact/` MUESTRA
M6	Build y despliegue	GitLab CI: build → test → despliegue a S3 detrás de CloudFront; assets con hash inmutables, HTML nunca cacheado	`.gitlab-ci.yml`
M7	Imágenes	Imágenes de hero y de tarjeta con variantes AVIF/WebP generadas en el build; el build falla si falta una variante	`scripts/generate-responsive-images.mjs`
M8	Puertas de calidad	ESLint, Jest (27 ficheros de test), Cypress y comprobación de imágenes y metadatos SEO antes del build	`scripts/check-*.mjs`, `src/__test__/`

3. Veredicto por módulo

Módulo	Veredicto	Por qué
M1 Enrutado e i18n	**Se conserva**	Una sola fuente de verdad para rutas y slugs; añadir una página son tres líneas. Se queda.
M2 SEO y prerender	**Se conserva**	El prerender es rápido y correcto. El cliente lo descartaba al cargar (ver B-01); resuelto hidratando en lugar de volver a renderizar.
M3 Pipeline del blog	**Se conserva**	Markdown con un parser de frontmatter por regex es frágil pero está documentado; un artículo mal formado desaparece del índice sin avisar, así que la puerta se mantiene.
M4 Analítica	**Se conserva, con una regla**	Los nombres de evento los vigila un test; la conversión que antes era un scroll (R-06) se sustituyó por eventos de contacto reales. Se mantiene la regla de solo herramientas propias.
M5 Contacto	**Se conserva**	Primero formulario, después WhatsApp, después calendario. Nada que reconstruir; se mantiene el interruptor de apagado fuera del dominio de producción.
M6 Build y despliegue	**Se conserva**	La política de caché es la correcta (assets inmutables, HTML sin caché). Se queda tal cual.
M7 Imágenes	**Reconstruido (hecho)**	El pipeline original servía un hero de 940 KB en la página de precios y 29 heros del blog de más de 300 KB sin variantes responsive (B-02). Reconstruido como paso del prebuild.
M8 Puertas de calidad	**Se conserva**	La suite se colgaba y por tanto no corría en la CI (R-03). Resuelto; ahora está en verde antes de cada build. El lint aún arrastra errores en componentes heredados (C-08).

4. Registro de incidencias, por gravedad

Gravedad: **bloquea** = impide el lanzamiento o cuesta clientes hoy · **riesgo** = fallará con usuarios o tráfico reales · **cosmético** = conviene arreglarlo, no conviene bloquear por ello.

ID	Gravedad	Módulo	Hallazgo	Causa	Solución	Estado
B-01	bloquea	M2	LCP en móvil por encima de 4 s en 17 URL (Search Console)	El cliente llamaba a `createRoot().render()`, descartaba el HTML prerenderizado y lo repintaba después de cargar tres bundles	`hydrateRoot` cuando el HTML coincide con la URL, con el chunk de la ruta precargado antes; los errores de hidratación se registran como evento	MUESTRA (CON-267, integrado el 12 de octubre). Volver a medir en Search Console a los 28 días
B-02	bloquea	M7	Hero de 940 KB en la página de precios y 29 heros del blog de más de 300 KB, servidos a tamaño completo en el móvil	Sin pipeline responsive; imágenes añadidas a mano	Variantes AVIF/WebP en el build con recorte cuadrado por debajo de 480 px, `<picture>` y preloads en el head; `check:images` falla si falta una variante	Resuelto (CON-269)
R-03	riesgo	M8	La ejecución de Jest nunca terminaba, así que la puerta de tests no corría en la CI	Un bucle infinito en el test de la página de error de ruta tras dejar un mock de analítica a medias	Bucle eliminado, metadatos del blog generados antes de la suite, suite ejecutada en la CI antes del build	Resuelto (CON-212, CON-213)
R-04	riesgo	M1	Un script de sincronización de traducciones sobrescribía todos los ficheros de idioma con 151 claves de una hoja de cálculo obsoleta	Herramienta heredada que siguió viva después de que la fuente de verdad pasara al repositorio	Script retirado con salida explicativa; los JSON de idioma son la única fuente; `check:locales`	Resuelto (CON-103)
R-05	riesgo	M2	Una página comercial sin indexar durante 11 semanas	El `lastmod` del sitemap no cambiaba nunca, así que el rastreador no tenía motivo para volver	`lastmod` real desde el historial de commits, un slug más limpio con 301 y enlaces internos desde los artículos relacionados	Resuelto (CON-216); indexada en cuatro días
R-06	riesgo	M4	Sesiones de pago partidas en dos en GA4 y «conversiones» que en realidad eran un scroll hasta el formulario	Doble page view al cargar; el evento de conversión saltaba al hacerse visible el panel	Un page view por carga; `lead`, `form_submit` y `whatsapp_click` como únicos eventos de contacto; HubSpot como fuente de verdad	Resuelto (CON-144, CON-166, CON-175)

ID	Gravedad	Módulo	Hallazgo	Causa	Solución	Estado
C-07	cosmético	M1	Cinco landings de tecnologías que el equipo no vende (Laravel, PHP, iOS, Angular, GraphQL)	Contenido escrito por volumen de búsqueda, no por el equipo real	Retiradas con redirecciones 301 a las páginas que sí existen	MUESTRA (CON-252)
C-08	cosmético	M8	ESLint reporta 14 errores en componentes heredados (fondo en canvas, roadmap, un formulario de contacto antiguo)	Anteriores a las reglas; sin cobertura de tests	Arreglar o eliminar los componentes; hacer el lint bloqueante en la CI	**Abierta**

5. Arquitectura objetivo

El producto no necesita una arquitectura nueva; necesita que la que tiene se respete bajo carga. El objetivo es la forma actual con tres reglas por escrito:

1. **Primero estático.** Toda página pública se prerenderiza en el build y se hidrata, nunca se vuelve a renderizar, en el cliente. Una página que no se puede prerenderizar es un bug, no una funcionalidad.
2. **Un solo pipeline para los assets.** Imágenes, tarjetas OG y variantes responsive se generan en el prebuild y se comprueban; nada entra en `public/` a mano.
3. **Medición solo con herramientas propias.** Los eventos se declaran en una lista, los vigila un test y están documentados en el plan de tracking. Los contactos se cuentan en HubSpot, no en la herramienta de analítica.

El stack se queda como está: React 18, TypeScript, Vite, React Router, i18next, un origen estático en S3 detrás de CloudFront y GitLab CI. Sin servidor que operar.

6. Plan de la siguiente fase

Ordenado para que cada tarea quepa en un desarrollador o en una sesión de agente, y para que las primeras paguen el resto.

1. Volver a medir Core Web Vitals tras B-01 y B-02 en las 17 URL; cerrar el ticket solo con datos de Search Console.
2. Dejar el lint (C-08) a cero y hacer `npm run lint` bloqueante en la CI.
3. Sustituir el parser de frontmatter por regex del blog por un parser YAML real con comprobación de esquema, manteniendo el orden de campos actual.
4. Añadir una ejecución smoke de Cypress sobre el build prerenderizado para los tres caminos de contacto (formulario, WhatsApp, calendario) en los tres idiomas.
5. Documentar el rollback del despliegue (el `dist/` anterior se conserva en el bucket durante una versión).

Presupuesto: en documento aparte.

7. Ficheros de contexto entregados

MUESTRA

- `CLAUDE.md` en la raíz del repositorio: stack, comandos, arquitectura, convenciones y la regla de que todo cambio nace de un ticket.
- Cuatro skills de proyecto para el agente de programación: sistema de diseño (solo tokens, un acento por pantalla), voz editorial y traducción, escritura anclada en datos y el flujo ticket → rama → merge request.
- El plan de tracking y la lista de eventos permitidos, vigilada por un test.
- Este documento, para que la siguiente persona (o agente) empiece desde el veredicto y no desde una suposición.

8. Qué no cubre este documento

- Pruebas de seguridad más allá de leer la configuración: no se ha hecho ningún test de penetración.
- Costes de hosting o de herramientas; este producto no tiene servidor y sus costes no son la cuestión aquí.
- Contenido y posicionamiento: es otra conversación.
- Nada del interior del producto de un cliente: esta muestra es nuestro propio código, elegido precisamente para que no aparezca nada tuyo.