

Cleexs — Descripción técnica y funcional (estado actual)

Fecha: 2026-02-04

Resumen ejecutivo

La solución Cleexs mide la visibilidad de una marca en respuestas de IA mediante corridas (runs) de prompts, parsea rankings Top 3, calcula PRIA agregado y expone resultados auditables por prompt. Incluye flujo de configuración de marca/competidores, generación de prompts, ejecución automática, métricas de análisis y sugerencias inteligentes de competidores.

Arquitectura general

Monorepo con tres capas principales: (1) API en Node/Fastify (apps/api), (2) Web en Next.js (apps/web) y (3) librería compartida (packages/shared). Persistencia en Postgres vía Prisma. Despliegue típico: API en Railway, Web en Vercel.

Componentes principales

API: Fastify + Prisma. Endpoints para tenants, marcas/competidores, prompts, runs y reportes. Web: Next.js con pantallas de Dashboard, Runs y Configuración. Shared: utilidades para scoring y normalización de marca.

Flujo funcional (end-to-end)

1) Configuración: se crea marca, se cargan competidores, se generan prompts desde el wizard y se crea un run. 2) Ejecución: el run ejecuta prompts en OpenAI, solicita Top 3 y persiste resultados. 3) Cálculo: se parsea el Top 3, se calcula score por prompt y PRIA agregado por run. 4) Visualización: Dashboard y Runs muestran ranking, métricas y detalle por prompt.

Modelo de datos (resumen)

Tenant, Plan, Brand, BrandAlias, Competitor, PromptVersion, Prompt, Run, PromptResult y PRIAReport. Los PromptResult almacenan respuesta completa, Top 3 parseado, score, flags y overrides. PRIAReport almacena PRIA total y PRIA por categoría.

Prompts y generación (V1 avanzada)

El wizard genera 10 prompts balanceados entre comparación, recomendación y defensibilidad. Todos piden Top 3 explícito e incluyen marca medida y competidores. Se muestra un score de calidad por prompt (formato Top 3 + mención de marca + competidores + longitud).

Sugerencia de competidores

La API expone un endpoint que usa OpenAI para sugerir 5–8 competidores directos con un motivo breve. Se filtran duplicados y se excluyen marcas ya cargadas. En la UI se muestran como chips clicables.

Parsing y scoring

Se parsea Top 3 con reglas de prioridad (lista numerada, bullets, párrafos). El score por prompt depende de la posición de la marca (1=1.0, 2=0.7, 3=0.4). PRIA total es el promedio de scores * 100. Se guardan flags de parsing para auditoría.

Métricas en Runs

En el detalle de un run se exponen métricas simples: confianza de formato (prompts con Top 3 parseable), tasa de mención de marca, tasa de aparición en Top 3 y tasa de posición #1. Se muestran tablas con resumen de apariciones por marca y posición media.

Endpoints principales

- GET /api/tenants/by-code/:code
- GET /api/brands?tenantId=
- POST /api/brands/:id/competitor-suggestions
- GET /api/prompts/prompt-versions?tenantId=
- POST /api/runs
- POST /api/runs/:id/execute
- GET /api/runs?tenantId=
- GET /api/reports/ranking?tenantId=
- GET /api/reports/pria?brandId=

Modelo de ejecución OpenAI

Las ejecuciones usan Chat Completions con el modelo gpt-4o-mini y un system prompt que exige Top 3 en formato numerado. Se envían la marca medida y los competidores para forzar comparación consistente.

Despliegue y configuración

API en Railway con variables: DATABASE_URL, OPENAI_API_KEY, API_URL, FRONTEND_URLS. Web en Vercel con NEXT_PUBLIC_API_URL apuntando al API. El deploy automático depende del webhook/branch configurado.

Estado actual (alcance real)

La solución implementa medición basada en Top 3, PRIA agregado, sugerencias de competidores con OpenAI, y métricas de visibilidad por run. No incluye todavía análisis de sentimiento, embeddings, ni scoring multicomponente.

Limitaciones conocidas

Los resultados dependen de la calidad del prompt y del modelo. Las métricas actuales no ponderan intención ni sentimiento. El PRIA se calcula por posición y no por valoración cualitativa. Se recomienda ampliar a V2 con métricas adicionales.