

Relevamiento inicial de RuleSet y activos críticos

Archivos de referencia analizados

Archivo	Descripción (resumen)	Evidencia clave
Blueprint V4 extendido (rw_b_blueprint_v_4_extendido_2025_08_06.md)	Documento central de la plataforma que define contexto, visión y las reglas universales para la versión 4. Contiene <i>metadatos YAML</i> completos (CODE, ID, VERSION, ROUTE, CROSSREF, AUTHOR, DATE) y establece principios como la Regla de máxima jerarquía (naming, ruta, workflow y OutputTemplate).	El front-matter muestra los campos obligatorios, con cross-references a la regla de compliance y al Master Plan ¹ . En la sección 2 destaca que el Glosario CODE v2 y el Diccionario CODE_TRIGGERS v2 son fuentes de verdad y de máxima jerarquía ² .
Master Plan V4 extendido (rw_b_master_plan_v_4_extendido_2025_08_06.md)	Define los objetivos y el contexto de la versión 4, enfatizando que la línea de base debe ser única y auditable. Destaca premisas como la obligatoriedad de naming/ruta/WF/ OutputTemplate, la ejecución de triggers y la documentación de <i>lessons learned</i> .	El front-matter incluye metadatos y cross-references al Blueprint y al prompt de baseline ³ . En las premisas se recalca que la omisión de los elementos obligatorios invalida cualquier activo y que los documentos de lookup (glosario y diccionario) son la fuente de verdad ⁴ .

Archivo	Descripción (resumen)	Evidencia clave
Glosario CODE v2 (rw_b_glosario_code_v_2_20250729.md)	Glosario de códigos con máxima jerarquía. Define la semántica universal: <code>CODE</code> en <code>SCREAMING_SNAKE</code> y <code>Name</code> en <code>PascalCase</code> , e incluye una tabla con identificadores y descripciones.	En la introducción se indica que el glosario tiene <i>máxima jerarquía</i> y que todo cambio debe reflejarse en el Diccionario de triggers ⁵ . Además, especifica instrucciones de uso, sincronización con el diccionario y reglas de semántica senior ⁶ .
Diccionario CODE_TRIGGERS v2 (rw_b_diccionario_code_triggers_v_2_20250729.md)	Contiene el listado de códigos, triggers y prompts asociados. Sirve como lookup rápido para scripts y prompts y debe estar alineado con el Glosario CODE v2.	El front-matter presenta los metadatos completos y las cross-references obligatorias ⁷ . Se indica que es un lookup ultrarrápido para prompts y scripts y que está 100 % alineado con el Glosario ⁸ .

Archivo	Descripción (resumen)	Evidencia clave
Prompt Codex Baseline V4 <p>(prompt_codex_baseline_v_4_check.md)</p>	Prompt de control para modelos, detalla los objetivos del barrido literal (LITW). Establece la necesidad de generar un baseline completo del árbol, detectar y crear carpetas/ README faltantes y estandarizar los README con metadatos y cross-references.	El front-matter incluye metadatos y cross-references, y la introducción remarca que todas las acciones deben acatar el Blueprint, Master Plan, Glosario y Diccionario ⁹ . Los objetivos listan la generación de un baseline detallado y la estandarización de todos los README ¹⁰ .

Observaciones y lineamientos

- Regla de máxima jerarquía:** Todos los documentos de referencia recalcan que ningún activo o flujo es válido si omite alguno de los cuatro elementos obligatorios: *naming*, *ruta exacta*, *workflow* y *OutputTemplate*. Esta regla se establece explícitamente en el Blueprint ² y se repite en el Master Plan ⁴.
- Precedencia del Glosario y Diccionario:** Tanto el Blueprint como el Master Plan indican que el **Glosario CODE v2** y el **Diccionario CODE_TRIGGERS v2** son la fuente de verdad para toda nomenclatura y triggers. Cualquier cambio en naming debe sincronizarse con estos lookups ² ⁶.
- Barrido literal (LITW):** El prompt de baseline define un barrido literal al 100 % del árbol para generar un baseline de carpetas y archivos, detectar faltantes y estandarizar los README ¹⁰. Este barrido no debe inferir ni modificar archivos, sólo recolectar información.
- Metadatos y cross-references:** Los documentos analizados presentan un bloque YAML inicial con campos estándar (CODE, ID, VERSION, ROUTE, CROSSREF, AUTHOR, DATE) y referencias cruzadas a Blueprint, Master Plan y los lookups. Esta estructura debe replicarse en todos los futuros README o workflows.
- Flujos y triggers:** Los documentos señalan la necesidad de incluir triggers ejecutables (como TRG_CONSOLIDATE_TL, TRG_AUDIT_TL, etc.) en cada ciclo y registrar lessons learned, changelog y checklists. Esta práctica asegura trazabilidad.

Próximos pasos sugeridos

- Ejecutar un barrido LITW sobre el repositorio:** Utilizar el script ops/scripts/litw_sweep.py o una herramienta equivalente para recorrer el 100 % del árbol (rglob), generando una tabla de baseline con las columnas sugeridas (ruta, metadatos, cumplimiento de

naming, presencia de OutputTemplate, etc.). Este barrido debe ser de lectura, sin modificar archivos.

2. **Comparar con la línea de base esperada:** El script `litw_sweep.py` utiliza el blueprint para calcular rutas esperadas y detectar faltantes, orphans o archivos legacy. Recomendamos adaptarlo para obtener un reporte actualizado, que luego servirá para actualizar `README.md`, `template_readme_v4.1` y para elaborar las versiones 4.1 del Master Plan y del Blueprint.
3. **Documentar resultados en Notion:** Crear páginas en Notion para visualizar flujos y diagramas (por ejemplo, flujos PDCA, matrices de compliance por plataforma), y almacenar los informes de baseline, incoherencias y lessons learned. Estas páginas serán la fuente operativa para futuras iteraciones.
4. **Actualizar archivos de referencia (sin modificar todavía):** Con el resultado del barrido se podrán identificar los README faltantes o incompletos. El siguiente paso, fuera de este relevamiento inicial, será actualizar `README.md`, `readme_template_v4.1.md`, el Master Plan 4.1 y el Blueprint 4.1, incorporando la jerarquía del Glosario y del Diccionario.

1 2 `rw_b_blueprint_v_4_extendido_2025_08_06.md`

https://github.com/gazton33/AingZ_Platform/blob/dc0f0ce87d17eca30c1b194d18b31a021a4eaa78/lifecycle/temp/rw_b_blueprint_v_4_extendido_2025_08_06.md

3 4 `rw_b_master_plan_v_4_extendido_2025_08_06.md`

https://github.com/gazton33/AingZ_Platform/blob/dc0f0ce87d17eca30c1b194d18b31a021a4eaa78/lifecycle/temp/rw_b_master_plan_v_4_extendido_2025_08_06.md

5 6 `rw_b_glosario_code_v_2_20250729.md`

https://github.com/gazton33/AingZ_Platform/blob/dc0f0ce87d17eca30c1b194d18b31a021a4eaa78/core/kns/glossary/rw_b_glosario_code_v_2_20250729.md

7 8 `rw_b_diccionario_code_triggers_v_2_20250729.md`

https://github.com/gazton33/AingZ_Platform/blob/dc0f0ce87d17eca30c1b194d18b31a021a4eaa78/core/data/dicts/rw_b_diccionario_code_triggers_v_2_20250729.md

9 10 `prompt_codex_baseline_v_4_check.md`

https://github.com/gazton33/AingZ_Platform/blob/dc0f0ce87d17eca30c1b194d18b31a021a4eaa78/lifecycle/temp/prompt_codex_baseline_v_4_check.md