

Informe de auditoría y propuesta de infraestructura RwB (Auditorías 1 & 2)

1 Workflow de auditoría RwB y roadmaps de consolidación

El flujo de auditoría v2 recoge las lecciones del ciclo piloto. Se apoya en **cuatro fases**:

1. **Barrido selectivo y mapeo estructural** – se extraen bloques, reglas y referencias clave del legado; cuando procede, se incluyen *snapshots* literales de matrices o tablas. Cada elemento se mapea al glosario y se sugiere la carpeta de destino ¹.
2. **Auditoría y consolidación estructural** – generación de outputs guía con estructura, checklist y referencias cruzadas; si existen redundancias se fusionan los archivos legacy y los originales quedan en carpetas separadas ².
3. **Feedback y memoria viva** – cada mejora o *lesson learned* se registra en la memoria viva (LEARN/INSI/CHGLOG); los outputs se enriquecen con feedback y rutas de referencia ³.
4. **Preparación para la integración global** – se marcan los bloques que necesitan validación con el legado para la migración final y se listan todos los archivos y lotes relacionados ⁴.

El workflow exige no eliminar el contenido original hasta finalizar la migración y propone una estructura mínima de carpetas (Legacy/Original, Auditorías, Consolidados, Legacy/Archive) ⁵. Un checklist acompaña cada lote de auditoría, verificando que se hayan realizado barrido, snapshot, references, lessons learned, separación de archivos y actualización de memoria ⁶.

Los **roadmaps de consolidado** guían la siguiente fase:

- El *mapeo de consolidado* para el Lote 2 agrupa diez archivos auditados en cuatro *clusters* ⁷:
- **Cluster A (Infraestructura & Onboarding)** combina el README raíz, `readme_base_aingz_t3_infra.md` y `readme_master_plan.md` para sintetizar onboarding, master plan y reglas de versión.
- **Cluster B (Matrices & Coverage)** unifica `readme_matriz_memorias_historiales.md`, `t3_raw_gold_matriz_final.md` y `readme_integracion_t2_memorias_historiales.md` en una única matriz maestra con checklist de cobertura ⁸.
- **Cluster C (Blueprint & Canvas)** consolida `t3_raw_gold_canvas_integrado_final_v2.md` y su diagrama de fases (testing local, automatización, integración multi-cloud, end-to-end y migración) ⁹.
- **Cluster D (Legacy & Migración)** resume `aing_z_repo_legacy_barrido_raw.md`, `aing_z_repo_raw_gold_scaffold.md` y `rawgold_scaffold_readme.md` para estandarizar la migración y reporting ¹⁰.
El mapeo sugiere outputs como un README consolidado de onboarding, una matriz maestra de coverage, un canvas/roadmap unificado y un documento de migración ¹¹.
- El **consolidado del blueprint/canvas** sintetiza el canvas, la metodología (análisis RAW → matriz modular → checklist de coverage → versionado → changelog → documentación cruzada) y las reglas de oro (nunca sobrescribir históricos; versionado obligatorio; onboarding a partir de

README/matriz/master plan; validar cobertura antes de migrar) ¹². Recomienda mantener sincronía entre canvas, matriz y master plan y actualizar las lecciones aprendidas en cada fase ¹³.

- El **roadmap de cierre y migración (Etapa 2b)** recuerda que antes de subir a GitHub se debe consolidar y auditar todo el material, definir una nueva estructura de carpetas y validar plantillas y workflows. Codex y GitHub son aceleradores, pero no sustituyen el control manual; los triggers de cierre sólo se ejecutan una vez que la nueva estructura está validada ¹⁴.

2 Análisis de las subcarpetas de Auditoría 1 y Auditoría 2

Auditoría 1 – Lote 1

El subdirectorio `Legacy/Aud_1/Legacy_Lote1/Auditorias/` contiene un único informe de consolidación (`rw_b_review_consolidacion_auditoria_lote_1_v_1_20250724.md`). Este documento presenta una matriz que evalúa cinco archivos legacy del lote 1 (documentos de onboarding, matrices de reglas, features y FAQs) e indica si son redundantes o deben mantenerse. A modo de ejemplo:

- `RwB_AUDT_MTXJERARQ_INSTR_LEGACY` y `RwB_AUDT_MTXINSTR_LEGACY` se consideran duplicados; se propone fusionar el diagrama y la tabla de precedencia en un único documento consolidado ¹⁵.
- `RwB_AUDT_MTXFEAT_PROMPTS_LEGACY` es una matriz de features que no está superada y se debe conservar en un subdirectorio de matrices ¹⁶.

El informe sugiere una estructura resultante para el lote 1 (README/ONBRD, FAQ/SOPORTE, DOCUMENTACIÓN, MATRICES/FEATURES y LEGACY/ARCHIVE) ¹⁷. Además recomienda versionar los archivos integrados, incluir la fecha del backup en el directorio `LEGACY/ARCHIVE` y automatizar los logs y checklists ¹⁸.

Auditoría 2 – Lote 2

En `Legacy/Aud_1/Legacy_Lote2/Auditorias/` se auditaron diez archivos clave. Cada informe sigue la plantilla estructurada que exige el workflow (identificación, propósito, barrido literal/mapping interno, compliance/naming, observaciones y checklist). Resumen de los hallazgos principales:

```
- **README raíz y `readme_base_aingz_t_3_infra.md`** - la matriz de auditoría de READMEs del Lote 2 confirma que la infraestructura modular, versionable y auditable está bien documentada. El README de base describe la infraestructura modular y automatizada; el README raíz actúa como eje de la IA/ChatGPT/OpenAI y ambos cumplen criterios de modularidad, versionado, trazabilidad y onboarding [519981444178727†L8-L14]. Las observaciones sugieren reforzar la actualización constante de la knowledge base y los enlaces cruzados en releases futuras [519981444178727†L29-L32].
- **`readme_integracion_t_2_memorias_historiales.md` y `readme_matriz_memorias_historiales.md`** - los informes de auditoría señalan que estos documentos centralizan la integración de memorias e historiales. `readme_integracion_t_2_memorias_historiales` define la problemática de la no-interoperabilidad entre hilos, explica la arquitectura para memorias e historiales resilientes, lista los entregables asociados
```

(matrices, plantillas, ejemplos, checklist) y detalla reglas de oro sobre versionado, feedback y naming incremental [86561355576149†L13-L24] .

`readme_matriz_memorias_historiales` documenta la matriz unificada de tipos de memoria e historial e incluye reglas de versionado, jerarquía y precedencia [151279375153470†L13-L23] . Las auditorías concluyen que ambos archivos cumplen portabilidad, autosuficiencia y reglas oro de versionado [86561355576149†L29-L36] .

- `**`readme_master_plan.md`**` - según la auditoría de READMEs, este archivo es el **entry point** del repositorio y referencia al **master plan**. Centraliza el onboarding, recomienda registrar lecciones aprendidas y prepara la integración futura [519981444178727†L10-L13] . No se detectan gaps críticos en su contenido ni en la referencia cruzada al master plan [519981444178727†L29-L32] .

- `**`t_3_raw_gold_canvas_integrado_final_v_2.md`**` - la auditoría reconoce que este canvas mapea la matriz RAW GOLD como blueprint principal para la gestión integral de carpetas, tipos de archivo, coverage y automatización. Presenta una metodología compuesta por análisis RAW, generación de matriz modular, checklist de coverage, versionado, changelog y documentación cruzada. Además resume un roadmap en cinco fases (testing local, automatización, integración multi-cloud, testing end-to-end y cierre/migración) e indica reglas de oro como versionado obligatorio y no sobrescribir históricos [653654146448422†L18-L24] .

- `t_3_raw_gold_matriz_final.md` - la matriz RAW GOLD final mapea componentes (knowledge, matrices, workflows, scripts, logs, docs, backups, notebooks, config y root), definiendo tipo, función y gaps. La auditoría solicita cargar ejemplos reales y automatizar logs/scripts ¹⁹ .
- `aing_z_repo_legacy_barrido_raw.md` - se valida el árbol literal de carpetas del ZIP legacy, identificando ausencia de carpetas estándar (`/knowledge/`, `/matrices/`, `/scripts/`, `/logs/` ...) y se recomiendan crear la infraestructura modular y generar READMEs y scripts de sync ²⁰ . - `aing_z_repo_raw_gold_scaffold.md` y `rawgold_scaffold_readme.md` - los informes de auditoría indican que estos archivos documentan la estructura jerárquica generada por el scaffold RAW GOLD (knowledge, matrices, workflows, scripts, logs, docs, config, backups, notebooks y onboarding) y explican que cada carpeta incluye un README y archivos demo como placeholders ²¹ . Se sugiere priorizar la sustitución de los archivos demo por plantillas reales y automatizaciones, y reforzar el logging y las pruebas de coverage ²² .

La **revisión cruzada** concluye que no hay duplicidades estructurales y que las recomendaciones son coherentes y pueden integrarse en la consolidación final. Las oportunidades de mejora convergen en la generación de plantillas base, la actualización incremental y el reporting automático de lecciones aprendidas ²³ .

Observaciones

- Todas las auditorías siguen una estructura homogénea y utilizan el glosario y las reglas de naming como referencia.
- El lote 2 evidencia madurez en la documentación (propósito claro, referencias cruzadas, reglas de oro), aunque faltan ejemplos concretos y scripts de automatización.
- El lote 1 se centró en un grupo pequeño de archivos legacy y priorizó la fusión de duplicados. Su review ya propone una estructura de directorios que puede tomarse como punto de partida para el diseño final.

3 Informe de auditoría propuesto para el paquete de Auditoría 2 (hardware/software)

No se localizaron auditorías para los **informes de inventario hardware/software** generados mediante el script `gz_n_56_vz_hw_info_script_full.py`. Este paquete comprende:

- `gz_n_56_vz_hw_info_script_full.py` – script que recopila información de CPU, RAM, almacenamiento, red, GPU y lista de software instalado y exporta un informe en CSV y Markdown ²⁴.
- `reporte_hw_gz_n56vz_2025-07-15_19-51.csv` – informe en CSV con datos detallados de hardware y software ²⁵.
- `reporte_hw_gz_n56vz_2025-07-15_19-51.md` – la misma información en formato Markdown ²⁶.
- `gz_n_56_vz_tech.md` – ficha técnica del portátil Asus N56VZ con observaciones de hardware, software instalado y enlaces a los informes generados ²⁷.

La auditoría propuesta debería contener:

1. **Identificación y metadatos** – nombrar cada archivo, indicar ubicación (adjuntos del proyecto), fecha de generación y versión.
2. **Propósito declarado** – el script y los informes sirven para crear un gemelo digital del equipo y documentar estado de hardware/software; son útiles para replicar entornos o realizar auditorías técnicas.
3. **Barrido literal y mapping** – describir los campos del informe (modelo de CPU, núcleos, RAM, discos, GPU, sistema operativo, IP, software instalado) y mapearlos al glosario (`CFG` para configuración, `LOG` para reportes, `DOC` para documentación técnica).
4. **Compliance y naming** – comprobar que los nombres siguen el template (prefijo del equipo, tipo de informe, fecha), que las rutas corresponden a `/knowledge/` o `/docs/` según su función y que se registran en `registro trazabilidad_total.md`.
5. **Observaciones y gaps** – sugerir incluir la versión del sistema operativo en el nombre del archivo, automatizar la subida a un directorio de `backups/` y registrar insights en la memoria viva.
6. **Checklist de auditoría** – verificar que el script exporta tanto `.csv` como `.md`, que el informe contiene todas las secciones, que los archivos se encuentran en el directorio correcto, y que el mapping al glosario (`CFG`, `LOG`, `DOC`, `BK`) está documentado.

Este informe completaría el paquete de Auditoría 2 y permitiría integrar los informes técnicos en la estructura de knowledge base y logs.

4 Propuesta de estructura de directorios para la infraestructura Rwb

El glosario core define códigos y categorías (Ruleset, Identificadores y Naming, Workflow, Knowledge, Log, Documentation, Script, etc.) ²⁸ ²⁹. Además, el checklist de clases de archivos resume las categorías encontradas en el legacy: matrices (MTR), scripts (SCR), workflows (WF), logs (LOG), documentación (DOC), knowledge (KNS), backups (BK), notebooks (NB), configuración (CFG), templates (TMP), checklists (CHK), imágenes (IMG) y planes/roadmaps (PLN) ³⁰. Integrando ambas taxonomías y las recomendaciones de los planes de migración ³¹ ³², se sugiere la siguiente estructura de carpetas:

```

/ (raíz del repositorio)
|
├─ universal/                # Versión activa y limpia del stack Rwb
|   └─ knowledge/            # Bases de conocimiento (KNS, MEM, NOTE, GLOS,
PREF, LEARN, INSI)
|       └─ kns_ctx/          # Contextos y líneas de base
|       └─ insights/         # Aprendizajes e insights
|       └─ glossary/         # Glosarios y diccionarios
|       └─ drafts/           # Drafts y snapshots de knowledge
|       └─ matrices/         # Matrices de precedencia, features y coverage
(MTR)
|   └─ workflows/            # Flujos y pipelines (WF); incluir scripts de
automatización específicos
|   └─ scripts/              # Scripts de automatización (SCR), subdivididos
en pipeline, test, integración, procesamiento y comandos 33
|   └─ logs/                 # Historiales y logs (LOG, BIT, CHG, TRC, ADT)
34
|   └─ docs/                 # Documentación final (DOC) con subcarpetas:
|       └─ rules/            # Reglas y normas del stack (Ruleset)
|       └─ plans/            # Roadmaps, master plan y planes de
implementación (PLN)
|       └─ checklists/       # Checklists y checkpoints (CHK, CHKP)
|       └─ glosario/         # Copia de glosarios oficiales
|       └─ templates/        # Plantillas oficiales (TMP, TEMPLG) y ejemplos
|       └─ templates/        # Plantillas universales de README (RDM),
prompts para IA (RDM_AI), templates humanos (RDM_H), matrices vacías, scripts
base...
|   └─ images/               # Diagramas y visualizaciones (IMG)
|   └─ notebooks/            # Notebooks y prototipos (NB)
|   └─ config/                # Archivos de configuración y variables de
entorno (CFG)
|   └─ backups/               # Copias de seguridad y snapshots (BK)
|   └─ onboarding/            # Guías y material de inducción (ONBG)
|   └─ feedback/              # Memorias de feedback y evaluaciones (FBCK,
EVLS)
|   └─ aux/                   # Otros módulos (e.g., tests, scripts
experimentales, extended tags)
|
├─ projects/                  # Repos específicos de proyectos (PR); cada uno
replica la estructura universal
|   └─ <project_name>/
|       └─ knowledge/
|       └─ matrices/
|       └─ workflows/
|       └─ ...
|
├─ clients/                   # Material para clientes (CL), misma estructura
que projects
└─ legacy/                    # Zona de staging y referencia de archivos

```

```

heredados
|   |— original/           # Archivos originales intactos
|   |— auditorias/        # Informes de auditoría por lote y paquete
|   |— consolidados/      # Consolidado de lotes (clusters A-D, matrices
maestras, canvas unificado...)
|   |— backups/           # Archivos legacy comprimidos una vez migrados
|   |— purgatorio/        # Zona de limpieza previa donde se trabaja la
migración
|— backup/                # Copias comprimidas de versiones anteriores; se
retira del repo una vez verificado

```

Principios clave de la estructura:

- **Modularidad:** cada carpeta tiene un *README* humano y otro para IA (RDM_H/RDM_AI), explicando su propósito, subcarpetas, naming y templates.
- **Versionado:** los nombres incluyen versión y fecha; al actualizar un archivo se genera un nuevo snapshot, nunca se sobrescribe el anterior ³⁵ ³⁶.
- **Trazabilidad:** todos los movimientos se registran en `registro_trazabilidad_total.md` y se reflejan en los logs/changelogs.
- **Glosario y naming:** cada nuevo término, carpeta o archivo debe validarse contra el glosario central y seguir los códigos definidos para Ruleset, identificadores/naming y workflow ²⁸. Las categorías (MTR, SCR, WF, LOG, DOC, KNS, BK, NB, CFG, TMP, CHK, IMG, PLN) guían la ubicación y el naming ³⁰.
- **Purgatorio y backup:** ninguna modificación del `legacy/` se integra al core hasta pasar por `purgatorio` y ser auditada; los originales se almacenan en `backups/` ⁵ ³⁷.

5 Alineamiento con los glosarios

El **glosario académico** define un árbol jerárquico de códigos (Ruleset, Identificadores/Naming, Workflow, Knowledge, Log, Documentation, Script, etc.) ²⁸ ³⁸. Estos códigos se usan para:

- **Nombrar archivos** (ej. `RWB_PR_DataAIEng_MTR_v1d2_FIN_t_3_raw_gold_matriz_final.md`) con el patrón `CTX_LYR_DOM_USC_MOD_CAT_TSK_NAME_STATE_VER_FECHA`; esto facilita la búsqueda y la clasificación automática.
- **Clasificar contenidos** en las carpetas propuestas: matrices (MTR), workflows (WF), scripts (SCR), logs (LOG), documentación (DOC), knowledge (KNS), backups (BK), notebooks (NB), configuración (CFG), templates (TMP), checklists (CHK), imágenes (IMG), planes (PLN) ³⁰.
- **Definir triggers y estados** (TRG, TYP, BK, MEM, NOTE, GLOS, etc.) que se reflejan en la metadata de los archivos ³⁹ ⁴⁰.

El glosario de *workflow* también aporta códigos como WF, MPLN, PLN, RMAP, CHK, CHKP, AUDT, TMPLG, LSWP, MIG, MAP y CLSS ³⁸. Estos se traducen directamente a carpetas y plantillas específicas en la estructura, asegurando que cada tipo de artefacto tenga un lugar definido (por ejemplo, `checklists/` para CHK/CHKP, `plans/` para PLN/RMAP, `auditorias/` para AUDT, `templates/` para TMPLG, `purgatorio/` para LSWP, `migration/` para MIG/MAP y `classification/` para CLSS).

Por último, los glosarios de **knowledge** y **log** establecen subcategorías como MEM, NOTE, LEARN, INSI, BIT, CHG y TRC ²⁹ ³⁴ . Estas subcategorías guían la organización interna de `/knowledge/` y `/logs/` y evitan mezclar notas rápidas con lecciones aprendidas o con el changelog.

6 Conclusiones y próximos pasos

- El workflow de auditoría define un proceso iterativo y trazable. Cada lote de auditoría produce informes homogéneos, lecciones aprendidas y outputs que se consolidan en clusters, listos para la siguiente fase de migración ⁴¹ ⁴² .
- Las auditorías de los Lotes 1 y 2 muestran madurez en documentación y versionado, aunque persisten gaps (falta de ejemplos, logging automático y plantillas de onboarding). La auditoría propuesta para los informes de hardware/software cubrirá otro paquete pendiente.
- La propuesta de infraestructura organiza el repositorio en torno a categorías del glosario (MTR, WF, SCR, LOG, DOC, KNS, BK, NB, CFG, TMP, CHK, IMG, PLN) y garantiza modularidad, versionado y trazabilidad. Integrar esta estructura es requisito previo para usar Codex y GitHub en la migración final ³¹ ³² .
- Seguir actualizando el glosario y las plantillas, documentar todas las migraciones en los registros de trazabilidad y emplear los scripts de mapping/backup permitirá acelerar las próximas auditorías y consolidaciones.

¹ ² ³ ⁴ ⁵ ⁶ ⁴¹ GitHub

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/workflow/rw_b_wf_auditoria_legacy_v_2_20250724.md

⁷ ⁸ ¹⁰ ¹¹ ⁴² GitHub

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_1/Legacy_Lote2/Auditorias/mapeo_consolidado_lote_2_v_1_20250725.md

⁹ ¹² ¹³ ³⁵ GitHub

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_1/Legacy_Lote2/Consolidados/consolidado_c_blueprint_canvas_lote_2_v_1_20250725.md

¹⁴ GitHub

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Learn/Relev_ideas/knowledge_draft_etapa_2_b_roadmap_cierre_pre_migracion_20250725.md

¹⁵ ¹⁶ ¹⁷ ¹⁸ GitHub

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_1/Legacy_Lote1/Auditorias/rw_b_review_consolidacion_auditoria_lote_1_v_1_20250724.md

¹⁹ ³⁶ GitHub

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_1/Legacy_Lote2/Auditorias/auditoria_t_3_raw_gold_matriz_final_v_1_20250725.md

²⁰ ³⁷ GitHub

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_1/Legacy_Lote2/Auditorias/auditoria_aing_z_repo_legacy_barrido_raw_v_1_20250725.md

²¹ ²² auditoria_aing_z_repo_raw_gold_scaffold_v_1_20250725.md

https://github.com/gazton33/RwB_Repo_AingZ/blob/ccaeccc46b0e1492fdcf406467432ced498ac8ff/Legacy/Aud_1/Legacy_Lote2/Auditorias/auditoria_aing_z_repo_raw_gold_scaffold_v_1_20250725.md

²³ revision_cruzada_auditorias_lote_2_v_1_20250725.md

https://github.com/gazton33/RwB_Repo_AingZ/blob/ccaeccc46b0e1492fdcf406467432ced498ac8ff/Legacy/Aud_1/Legacy_Lote2/Auditorias/revision_cruzada_auditorias_lote_2_v_1_20250725.md

24 **GitHub**

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_3/Legacy_Original/Lote_Inventario/gz_n_56_vz_hw_info_script_full.py

25 **GitHub**

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_3/Legacy_Original/Lote_Inventario/reporte_hw_gz_n56vz_2025-07-15_19-51.csv

26 **GitHub**

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_3/Legacy_Original/Lote_Inventario/reporte_hw_gz_n56vz_2025-07-15_19-51.md

27 **GitHub**

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_3/Legacy_Original/Lote_Inventario/gz_n_56_vz_tech.md

28 29 33 34 38 39 40 **GitHub**

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/knownledges/glossary/rw_b_glosario_code_v_0_core_academico.md

30 **GitHub**

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_4/Legacy_Original/Lote_docs/checklists/checklist_clases_archivos.md

31 **GitHub**

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_4/Legacy_Original/Lote_docs/plans/baseline_roadmap.md

32 **GitHub**

https://github.com/gazton33/RwB_Repo_AingZ/blob/main/Legacy/Aud_4/Legacy_Original/Lote_docs/procedimiento_migracion.md