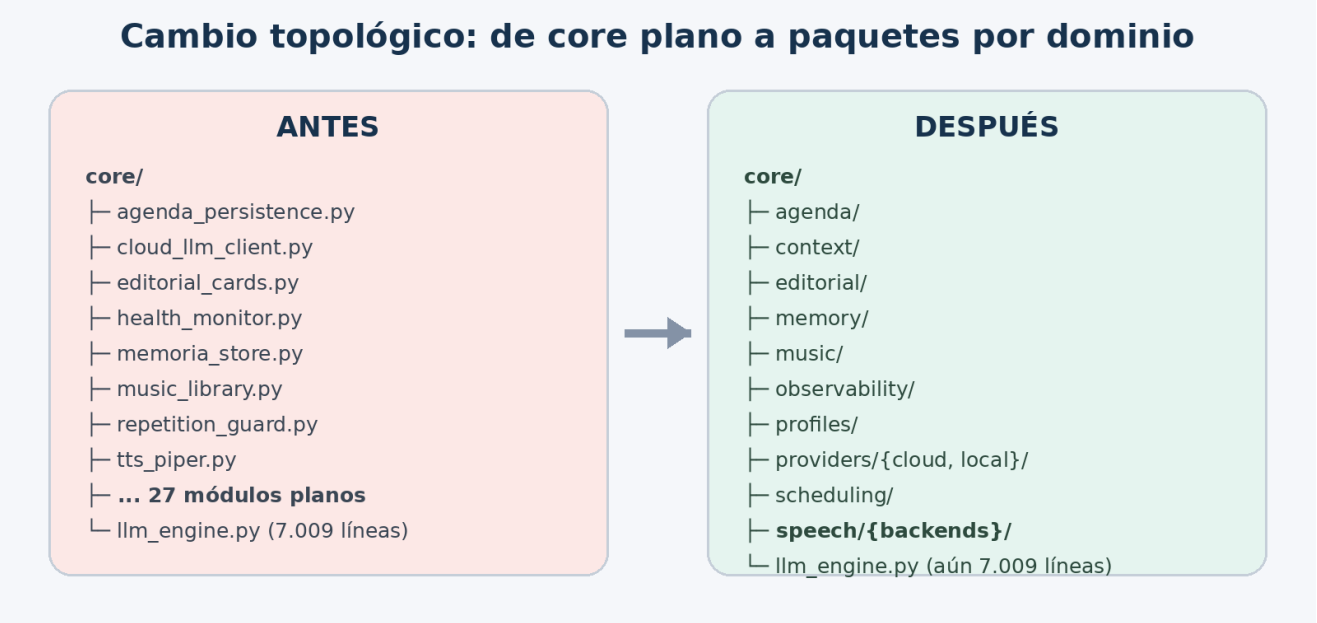
OpenCohost

Migración de `core/` a paquetes por dominio

Informe técnico de ejecución, incidentes silenciosos, validación y lecciones operativas

27	150	5.129
MÓDULOS REUBICADOS	REFERENCIAS OBSOLETAS	TESTS VERDES FINALES



Documento sintetizado a partir del reporte de Claude compartido por el propietario. No sustituye una auditoría independiente del repositorio.

Estado documentado: commit e0cebb3 | Rama refactor/llm-engine-split-20260802 | 2 de agosto de 2026

1. Resumen ejecutivo

La migración no fue un cambio cosmético: expuso acoplamientos físicos que el sistema mantenía ocultos.

✓

Veredicto

La reorganización por dominios quedó completada y commiteada con la suite verde. Sin embargo, el proceso demostró que mover archivos en Python puede alterar comportamiento aun cuando los imports compilen y la suite pase.

45	175	25	14
ARCHIVOS CON RUTAS MUERTAS	RUTAS DEL CANDIDATO REVISADO	LÍNEAS DE CORRECCIÓN	SKIPS, SIN CRECER

Resultado consolidado

- El propietario movió manualmente 27 módulos desde un `core/` plano hacia subpaquetes por dominio.
- La verificación determinista detectó 150 referencias antiguas distribuidas en 45 archivos.
- Los puntos de entrada, `compileall` y la colección de 5.142 tests quedaron limpios tras el codemod.
- La suite completa encontró una referencia por ruta de disco que las búsquedas de imports no podían detectar.
- Una revisión de cuatro lentes descubrió dos defectos runtime adicionales: una ruta derivada de `__file__` y un import tardío reordenado.
- El candidato final cerró en 5.129 tests aprobados y 14 omitidos, incluyendo un nuevo test de regresión.

Estado actual

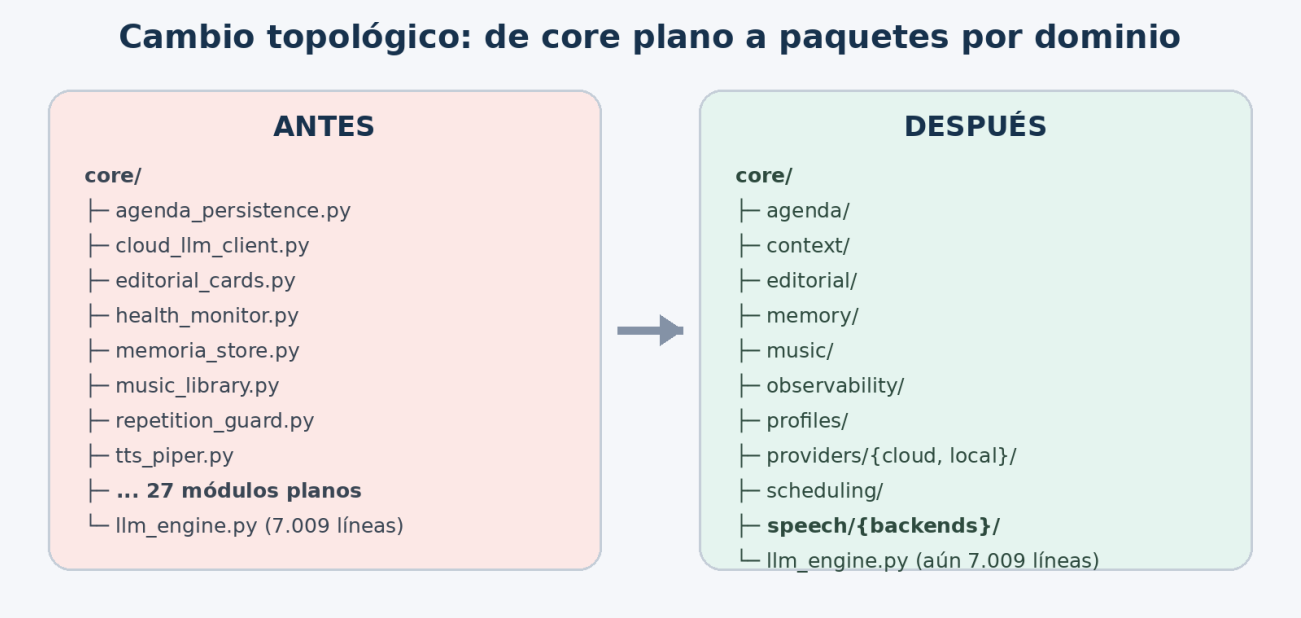
!

La reorganización terminó; el split de `llm_engine.py` todavía no empezó.

El archivo principal conserva 7.009 líneas. La migración por dominios es la base estable sobre la que debe ejecutarse después el plan owner-driven de mixins.

2. Alcance real de la migración

Lo que parecía un movimiento de carpetas terminó siendo una migración transversal del grafo de módulos.



La nueva topología mejora navegación, límites de responsabilidad y costo futuro de mantenimiento.

Qué cambió

Elemento	Antes	Después
Topología	`core/` plano con 27 módulos	Subpaquetes por agenda, memoria, speech, providers, etc.
Imports	Rutas planas `opencohost.core.x`	Rutas de dominio `opencohost.core.<dominio>.x`
Riesgo oculto	Dependencias físicas invisibles	Acoplamientos revelados y documentados
Suite	Piso 5.128 / 14	5.129 / 14 con test de regresión
Motor principal	7.009 líneas	Sigue en 7.009 líneas; split pendiente

Qué no cambió

- No se pretendía rediseñar comportamiento de producto.
- Los 21 renames completamente mecánicos conservaron blobs idénticos.
- Los otros seis renames fueron auditados como cambios de imports y una línea de docstring.
- La reorganización no incluyó todavía la extracción de métodos de `MotorVocalIA`.

Lectura arquitectónica

La organización por dominios no es solo estética. Reduce la probabilidad de que una nueva extracción vuelva a producir archivos provisionales en la raíz y obliga a declarar dónde pertenece cada responsabilidad.

3. Secuencia de ejecución y validación

La parte manual decidió el dominio; la parte automatizada actuó como notario.

La secuencia final combinó criterio humano, codemod determinista, tests y revisión semántica.

Fase	Objetivo	Evidencia producida
Mapa por dominios	Asignar cada módulo a una responsabilidad	Nueva estructura de `core/`
Movimiento manual	Mantener control del alcance	27 renames detectados por Git
Codemod	Actualizar rutas antiguas sin recorrer worktrees ajenos	150 sustituciones uno a uno
Importación	Verificar puntos de entrada y sintaxis	6 entrypoints + `compileall`
Colección	Cazar imports rotos en tests	5.142 tests, 0 errores
Suite completa	Ejecutar rutas que solo fallan en runtime	1 guard test de path físico
Revisión 4R	Detectar riesgo, legibilidad, fiabilidad y resiliencia	2 defectos runtime + 1 hallazgo refutado
Corrección	Reparar sin ampliar el alcance	25 líneas + test
Commit	Cerrar en estado revertible y verde	`e0cebb3`, 5.129 / 14

!

Regla de alcance aprendida

El codemod debe operar sobre `git ls-files`, no sobre todo el disco. El primer intento habría entrado en `.orchestrator/worktrees/` y podía modificar árboles de trabajo fuera del candidato.

4. Primer hallazgo: el árbol no importaba

La extensión de VS Code movió archivos y actualizó una parte de las referencias, pero dejó rutas colgando.

150	45	0
REFERENCIAS MUERTAS	ARCHIVOS AFECTADOS	ERRORES TRAS EL CODEMOD

Causa

- `llm_engine.py` y otros importadores seguían usando rutas planas.
- Los subpaquetes no ofrecían reexports que hicieran válidas esas rutas antiguas.
- La UI legacy, la API y el motor dependían del mismo grafo de importación.

Corrección determinista

```
# Principio aplicado
scope = git ls-files

for cada ruta antigua:
    reemplazar solo referencias rastreadas
    preservar símbolos y comportamiento
    no tocar worktrees, temporales ni copias externas
```

Por qué no bastaba revisar a ojo

Método	Fortaleza	Límite
Tour manual	Detecta anomalías de contexto	No escala a cientos de archivos
Pylance / VS Code	Actualiza referencias simples	Puede omitir strings, tests y rutas dinámicas
`rg`	Encuentra patrones conocidos	No descubre semántica derivada de posición
Codemod	Repetible y auditable	Debe limitarse al conjunto correcto
Suite	Ejecuta contratos reales	Puede tener mocks que oculten rutas físicas

Conclusión

Desconfiar de la extensión fue correcto. El error no fue mover manualmente; fue confiar inicialmente en que una herramienta de editor había cubierto todo el repositorio.

5. Tres defectos silenciosos introducidos por el movimiento

Los errores más importantes no eran simples imports.

Defecto	Por qué ocurrió	Qué lo detectó	Corrección
Guard test con ruta física	El test leía `core/audio_bed.py` desde disco	Suite completa	Actualizar el path al dominio `music/`
`SERVER_SCRIPT` un nivel corto	La fórmula con `dirname(__file__)` asumía la ubicación antigua	Ejecutar resolución + lentes R1/R3	Subir un nivel y añadir test de existencia
Import tardío de `qwen_tts` hoisted	El ordenador movió el import antes del setup de variables HF	Diff de orden + lente R4	Restaurar import tardío y documentar `noqa`

5.1 Ruta derivada de `__file__`

```
# Antes del movimiento: dos niveles llegaban a opencohost/
core/health_monitor.py
    dirname(dirname(__file__)) -> opencohost/

# Después del movimiento: faltaba un nivel
core/observability/health_monitor.py
    dirname(dirname(__file__)) -> opencohost/core/  # incorrecto
```

La suite no lo detectó porque los tests mockeaban `subprocess.Popen`; el argumento nunca se validaba contra el sistema de archivos.

5.2 Orden de imports con semántica

```
# Orden requerido
configurar HF_HOME / HF_HUB_CACHE / HF_HUB_OFFLINE
from qwen_tts import Qwen3TTSModel  # import tardío deliberado

# Orden roto por el sorter
from qwen_tts import Qwen3TTSModel
configurar variables  # demasiado tarde
```

!

Principio

El orden de imports no siempre es inerte. Si una dependencia lee variables de entorno durante su importación, reordenar el bloque modifica comportamiento runtime.

6. Evidencia de verificación

La suite fue necesaria, pero no suficiente; la revisión semántica cerró los huecos.

5.142	6	4	5.129
TESTS RECOLECTADOS	PUNTOS DE ENTRADA	LENTES DE REVISIÓN	PASSED FINALES

Control	Resultado	Qué demuestra
Git rename audit	21 blobs idénticos; 6 auditados hunk por hunk	La mayoría de movimientos no alteró contenido
Imports de entrypoints	Motor, engine_host, API, CTK, `__main__`, editorial CLI	El grafo principal carga
`compileall`	Limpio	No hay sintaxis rota
Colección de pytest	5.142, cero errores	No se perdieron ni duplicaron tests
Suite inicial	5.127 / 14 / 1 fallo	Descubrió el test por ruta física
Suite tras primer arreglo	5.128 / 14	Recuperó el baseline
Revisión 4R	Riesgo alto; defectos runtime	La suite verde no equivalía a migración segura
Suite final	5.129 / 14	Baseline + nuevo test de regresión

Qué aportó cada lente

Lente	Aporte
R1 - Riesgo	Clasificó `SERVER_SCRIPT` como crítico e introducido.
R2 - Legibilidad	Señaló deuda no bloqueante: `profiles/profiles.py` y docstrings obsoletos.
R3 - Fiabilidad	Corroboró el bug y verificó que monkeypatches y colisiones no fueran defectos vivos.
R4 - Resiliencia	Detectó el import tardío roto; otro hallazgo fue refutado diferencialmente como preexistente.

Disciplina de evidencia

Un hallazgo no se aceptó solo porque lo reportara un agente. Se comparó con la versión anterior y se refutó cuando fallaba igual antes y después.

7. Checklist para futuras migraciones de archivos

La próxima reorganización debe verificarse en minutos, no con otro tour manual por cientos de módulos.

Capa	Búsqueda o prueba	Defecto que cubre
Imports	Rutas punteadas antiguas y `from ... import ...`	ImportError
Strings	`patch`, `monkeypatch`, `import_module`	Referencias dinámicas
Filesystem	`"core" / "x.py"`, `Path(...)`	Tests y scripts por ruta física
Posición	`__file__`, `parents[`, `dirname`	Rutas relativas al módulo
Orden	Diff de bloques de imports contra HEAD	Side effects en import time
Empaquetado	Hatch, PyInstaller, entrypoints	Archivos ausentes en build
Sintaxis	`compileall`	Errores de parseo
Descubrimiento	`pytest --collect-only`	Imports rotos en tests
Comportamiento	Suite completa	Rutas ejecutadas tardíamente
Runtime	Smoke tests de procesos, TTS y recursos	Mocks que ocultan dependencias reales

Protocolo recomendado

1. Congelar el mapa viejo → nuevo antes de editar.
2. Mover un conjunto coherente y limitar el codemod a `git ls-files`.
3. Revisar el diff de renames y el orden de imports.
4. Buscar rutas punteadas, strings, paths físicos y derivaciones de `__file__`.
5. Importar todos los puntos de entrada y ejecutar `compileall`.
6. Recolectar tests; luego ejecutar la suite completa.
7. Ejecutar smoke tests de límites de proceso y archivos externos.
8. Aplicar correcciones acotadas, añadir regresiones y recontar la suite.
9. Commitear solo con árbol limpio y evidencia registrable.

No repetir

No volver a revisar 600 archivos uno por uno. El criterio humano decide el dominio; las herramientas mecánicas deben verificar exhaustividad.

8. Estado cerrado y siguiente frontera

La migración creó una base más ordenada; no reemplazó el trabajo pendiente sobre el God Object.

e0cebb3	5.129 / 14	7.009
COMMIT DE MIGRACIÓN	PISO PARA EL SPLIT	LÍNEAS EN LLM_ENGINE

Cerrado

- 27 módulos trasladados a paquetes de dominio.
- 150 referencias antiguas corregidas.
- Tres defectos introducidos detectados y corregidos.
- Suite final verde y nuevo test de regresión.
- Commit creado; árbol reportado como limpio y sin push.

Pendiente no bloqueante

- Resolver el nombre redundante `profiles/profiles.py` en un commit separado.
- Actualizar referencias obsoletas en comentarios y documentación.
- Decidir el destino del directorio vacío `core/scout/`.
- Mantener o añadir `__init__.py` como decisión explícita, aunque el empaquetado actual no los requiera.

Siguiente trabajo principal

→

Iniciar el Paso 1 del split de `llm_engine.py`.

La reorganización no extrajo ningún mixin. El archivo sigue en 7.009 líneas; ahora puede dividirse directamente hacia los dominios definitivos con una base de 5.129 tests aprobados y 14 omitidos.

Conclusión técnica

La migración fue correcta, pero el valor principal no fue solo obtener carpetas bonitas. El proceso convirtió dependencias físicas implícitas en contratos verificables: rutas derivadas, orden de imports, paths leídos por tests y límites de proceso. La próxima extracción partirá de una topología más clara y de una checklist que reduce el costo de volver a mover código.

Anexo A. Registro ejecutivo de la decisión

Resumen listo para ADR, handoff o descripción de commit.

```
Decisión
Reorganizar opencohost/core desde una topología plana hacia paquetes por dominio
antes de iniciar el split de llm_engine.py.

Razón
Evitar que los nuevos mixins nazcan en una ubicación provisional y reducir el
costo de navegación, ownership y futuros movimientos.

Ejecución
- 27 módulos trasladados manualmente.
- 150 referencias obsoletas corregidas con codemod acotado a git ls-files.
- Entry points, compileall, colección y suite completa validados.
- Revisión 4R sobre candidato de riesgo alto.
- Corrección acotada de defectos runtime y nuevo test de regresión.

Resultado
Commit e0cebb3.
Suite: 5.129 passed / 14 skipped.
Sin regresiones conocidas en el alcance validado.

Advertencia
Una suite verde no cubre automáticamente rutas derivadas de __file__, orden
semántico de imports ni tests que leen archivos por path físico.

Siguiente paso
Ejecutar el split owner-driven de llm_engine.py sobre este nuevo baseline.
```

Fuente y límites

- El contenido sintetiza el registro de trabajo y la conclusión final proporcionados por Claude.
- El reporte contiene repeticiones y fragmentos truncados propios de una sesión larga; este documento conserva los resultados consistentes de la fase final.
- No se inspeccionó aquí el commit `e0cebb3` ni el repositorio vivo; las cifras se presentan como evidencia reportada en la fuente.

Frase de cierre

La organización por dominios no es cosmética: hace que el próximo movimiento sea barato, verificable y reversible.