

Acerca de esta wiki

Notas técnicas desde casa

Un espacio para documentar infraestructura, servicios, automatizaciones y decisiones técnicas que terminan formando parte del stack.

Esto existe por una razón bastante simple: **cansancio de olvidar cómo se arregló algo que ya funcionó una vez.**

Aquí no hay teoría de manual ni artículos escritos para rellenar. Lo que hay son configuraciones reales, pruebas, errores, integraciones y soluciones que han salido de pelearse con servicios funcionando en producción doméstica.

“ **Idea principal:** esta wiki funciona como memoria extendida. Sirve para recordar cómo quedó montado algo, qué problemas aparecieron, qué decisiones se tomaron y por qué terminó funcionando así.

☐ Qué es esto

Una colección organizada de apuntes técnicos sobre homelab, Docker, redes, seguridad, sistemas Linux, automatización, observabilidad y servicios autohospedados.

El objetivo no es únicamente guardar configuraciones, sino también conservar el contexto: qué se probó, qué alternativas se descartaron y cómo encaja cada componente dentro del resto de la infraestructura.

Muchas entradas son documentación estable. Otras son pruebas, notas rápidas, integraciones específicas o pequeños hallazgos que merece la pena no perder.

☐ Cómo está organizada

1. **Estantes por temática**: contenedores, seguridad, automatización, redes, monitorización, sistemas y servicios.
2. **Búsqueda global** para localizar rápidamente herramientas, configuraciones o integraciones.
3. Contenido agrupado por servicio, integración, flujo, problema resuelto o decisión técnica.

Cuando algo no encaja bien dentro de una jerarquía estricta, termina en secciones más abiertas como *Notas técnicas* o *Tips útiles*. Ahí suelen acabar ajustes rápidos, pruebas, validaciones o soluciones concretas que merece la pena conservar.

Algunos servicios aparecen en varios estantes cuando tiene sentido tratarlos desde perspectivas distintas: despliegue, hardening, autenticación, mantenimiento o automatización.

☐ Qué vas a encontrar

- Configuraciones reales y referencias a repositorios.
- Integraciones entre servicios del stack.
- Decisiones técnicas explicadas y alternativas descartadas.
- Problemas reales con soluciones prácticas.
- Scripts y automatizaciones para reducir trabajo repetitivo.
- Ajustes de seguridad aplicados sobre servicios internos y expuestos.
- Notas experimentales y documentación más informal cuando algo merece quedar registrado.

En resumen: contenido basado en uso real y mantenimiento continuo, no ejemplos bonitos montados únicamente para una captura.

☐ Código y configuraciones

Gran parte de las configuraciones y archivos relacionados con esta wiki están disponibles en mi [Gitea personal](#).

Ahí están los `docker-compose.yml`, configuraciones de Caddy, scripts, automatizaciones y otros recursos asociados a muchos de los artículos.

La idea es mantener separadas ambas partes:

- La explicación, decisiones y contexto aquí.
 - Los archivos reales y el versionado donde corresponde.
-

☐☐ Filosofía

Aprender · Probar · Romper · Documentar

Esta wiki empezó como un conjunto de notas sueltas y terminó convirtiéndose en una bitácora técnica bastante más estructurada.

No pretende ser documentación académica ni una colección de tutoriales genéricos. La prioridad es dejar constancia de cómo están montadas las cosas, qué problemas aparecieron y qué soluciones terminaron mereciendo la pena.

Porque cuando pasan seis meses y algo deja de funcionar, la memoria humana tiene aproximadamente la misma fiabilidad que un contenedor marcado como `latest` un viernes por la noche.

⚠ Aviso rápido

- El contenido está probado en mi entorno.
 - Cada infraestructura tiene limitaciones, dependencias y rarezas distintas.
 - Nada debería aplicarse a ciegas sin validar antes.
-

☐☐ Contacto

- [Página de servicios](#)
 - [LinkedIn: Juan Francisco Trapero](#)
-

☐☐ Agradecimientos

- **BookStack** por servir de base para organizar toda la documentación.
 - **La comunidad open-source** por compartir herramientas, ideas y tiempo.
-

☐☐ Otras voces técnicas

Espacios con contenido técnico que merece la pena seguir por la calidad de lo que publican.

- [Unravel Engineering](#) – Programación, algoritmia, física, criptografía y computación gráfica con un enfoque técnico bastante sólido.

Revision #31

Created 2024-11-19 09:03:00 UTC by Juan Francisco

Updated 2026-05-15 12:37:09 UTC by Juan Francisco