SISTEMAS + IA / TALLER NORTE

Cloud para sistemas que siguen funcionando

Publicá, observá y recuperá aplicaciones, bases y automatizaciones.



Con proyecto de referencia y pruebas reproducibles

EDICIÓN VISUAL 02 · NOVIEMBRE 2026

Qué vas a construir y cómo usar la guía

La referencia es Taller Norte, un negocio ficticio de servicios. Sus consultas, responsables y pedidos permiten observar un proceso completo sin utilizar información privada. Las capturas provienen de la aplicación local incluida en el paquete.



Figura de referencia · El tablero del laboratorio muestra 3 abiertas, 1 vencida y 1 para hoy con los datos iniciales.

El laboratorio simula eventos y pagos. No se conecta a Meta, WhatsApp, Mercado Pago ni servicios de IA. Para esos destinos, la guía identifica las comprobaciones y permisos que se agregan. Los prompts de Claude son ejemplos preparados para el lector, no transcripciones de una sesión ejecutada.

Antes de abrir las herramientas

Un inventario técnico, un plan de despliegue y una prueba de restauración.

Perfil y requisitos

Desarrolladores que van a operar sistemas para clientes y necesitan una rutina de mantenimiento.

Un entorno de pruebas y acceso autorizado al hosting, DNS y base.



Ilustración editorial · La imagen introduce el proceso. Las pantallas de las páginas siguientes muestran el ejercicio que lo implementa.

MUESTRA · NO ES EL PRODUCTO COMPLETO

Dibujar lo que realmente hay que operar

Listá aplicación web, backend, n8n, base, correo, almacenamiento y proveedores. Indicá quién los administra, dónde viven y qué depende de cada uno. El diagrama debe mostrar entrada, persistencia y acciones externas para entender qué pasa si una parte deja de responder.

Definí qué datos necesitan persistir y cuáles se pueden reconstruir. El código puede reinstalarse desde una versión conocida, pero las credenciales y la base requieren procedimientos propios. En proyectos compartidos, inventariá nombres y permisos para evitar que una migración afecte a otra aplicación. Usá prefijos y un esquema propio cuando corresponda.

DIBUJAR LO QUE REALMENTE HAY QUE OPERAR



SECUENCIA DEL CASO · cada etapa tiene una entrada y una comprobación

REGLA DEL CASO

Componente = función + dueño + datos + dependencia

Anotá qué sucede si cae y qué puede seguir operando. Una página estática puede permanecer disponible mientras la base se recupera, pero el checkout debe informar su estado correctamente.

MUESTRA · NO ES EL PRODUCTO COMPLETO

Dibujar la dependencia que hace visible cada pantalla

El tablero depende de aplicación y base local. Una pantalla disponible no garantiza que haya leído o guardado los datos correctos.



MUESTRA · NO ES EL PRODUCTO COMPLETO

Captura del proyecto de referencia · Datos ficticios del laboratorio local.

Acción	Entrada	Resultado esperado
Identificar entrada	Navegador	Se conoce el cliente
Identificar cálculo	dominio.mjs	Se ubica regla
Identificar persistencia	SQLite local	Se conoce dónde queda el cambio

Resolver el ejercicio

Referencia para implementar

```
navegador -> servidor.mjs
servidor -> dominio.mjs
servidor -> datos/laboratorio.sqlite
proveedores externos: simulados
```

Este corte pequeño permite aprender operación sin desplegar cuentas públicas. Cuando cambies la base, verificá consultas, migraciones, permisos y copia de archivos. El diagrama de dependencias cambia con ese movimiento.

Probalo con una variante

¿Qué se pierde si copiás solo servidor.mjs?

SOLUCIÓN COMENTADA

La lógica importada, el estilo, las pruebas y los datos persistidos. La entrega necesita el conjunto de dependencias y una forma de reconstruirlo.

Registrá la evidencia

Guardá entrada, salida observada y diferencia respecto al criterio. Si falla, identificá el primer paso que cambia los datos y repetí la prueba después de corregirlo.

MUESTRA · NO ES EL PRODUCTO COMPLETO