EcomLabs

PLANTILLA Y CHECKLIST

Plan de rollback antes de desplegar

La plantilla del plan de rollback en siete líneas, los tres pasos de expandir y contraer para que el código anterior siga funcionando, qué verificar después del deploy y la checklist final. El SQL y el script de verificación están probados en el artículo.

PARA	Equipos que despliegan una app web con base de datos PostgreSQL
PROBADO CON	Node 22.22.3 · PGlite 0.5.10 (PostgreSQL 18.6) · Windows 11
ACTUALIZADA	12 de noviembre de 2026
DOCUMENTACIÓN	railway.com/guides/roll-back-bad-deploy
ARTÍCULO	ecomlabs.dev/g/rollback
AUTOR	Luis Morales · @luismoralesdev

Índice

Antes de empezar	Qué cubre esta guía
Paso 01	La plantilla del plan de rollback
Paso 02	Migrar sin romper la vuelta atrás
Paso 03	Verificar después de desplegar
Al terminar	Checklist después del deploy

ANTES DE EMPEZAR

Qué cubre esta guía

Un plan de rollback es la nota que escribes antes de desplegar para que volver atrás sea un paso decidido y no una discusión. Aquí está la plantilla, cómo migrar sin romper el código anterior, qué verificar después y la checklist final. Probamos el SQL en PGlite 0.5.10 (PostgreSQL 18.6 en WebAssembly) y el script de verificación contra un servidor local de prueba, con Node 22.22.3. Los pasos de la plataforma salen de la documentación de Railway y no se ejecutaron en un despliegue real.

PASO 01

La plantilla del plan de rollback

Llénala una vez por servicio, antes del deploy. Los ejemplos son ilustrativos.

CAMPO	QUÉ ESCRIBIR	EJEMPLO ILUSTRATIVO
Versión nueva	El identificador que devuelve `/healthz`	`a1b2c3d`
Versión a la que vuelves	El último identificador que funcionó, no «el anterior»	`9f8e7d6`
Cómo vuelves	Pasos exactos y quién tiene acceso	Deployments, deployment `9f8e7d6`, menú de tres puntos, Rollback
Ventana	Cuánto guarda la plataforma la imagen anterior	72 h (plan Hobby)
Base de datos	Última migración y si ya se contrajo	Expandida, sin contraer
Criterio	Qué resultado dispara el rollback	El script de verificación sale con código 1
Quién decide	Un nombre, con teléfono	Responsable técnico
A quién se avisa	Ventas, soporte, la agencia	Canal del equipo

Según la guía de Railway, un rollback restaura la imagen Docker y las variables personalizadas como un deployment nuevo, sin reconstruir. La retención de la imagen es de 24 horas en Free/Trial, 72 en Hobby, 120 en Pro y 360 en Enterprise; después queda Redeploy, que reconstruye desde el código original. Esa guía no dice nada de bases de datos ni migraciones.

PASO 02

Migrar sin romper la vuelta atrás

Un `RENAME COLUMN name TO full_name` en un solo paso rompió al código anterior en nuestra prueba (`column "name" does not exist`). Con expandir y contraer, cada paso funciona con las dos versiones:

PASO	CUÁNDO	QUÉ HACES	¿VUELTA A LA VERSIÓN ANTERIOR?
1 · Expandir	Antes del deploy	`ADD COLUMN full_name text`, un trigger que sincroniza con `name` y un `UPDATE` de relleno	Sí
2 · Migrar	En el deploy	La versión nueva lee y escribe solo `full_name`	Sí
3 · Contraer	Pasada la ventana del rollback	`SET NOT NULL`, `DROP TRIGGER`, `DROP FUNCTION`, `DROP COLUMN name`	No

> **CUIDADO**
> El SQL completo del trigger y la prueba están en el artículo. Contraer cierra la vuelta atrás, así que anótalo en el plan. En una tabla grande, haz el relleno por lotes y prueba el trigger con tus propios escritores.

PASO 03

Verificar después de desplegar

El healthcheck de la plataforma corre solo al desplegar y no vigila después. Verifica tú, con un script que salga con código distinto de 0 si algo falla (el script completo está en el artículo). Como mínimo comprueba:

- `/healthz` devuelve la versión que acabas de desplegar (detecta que sigue sirviendo la vieja).
- La portada responde 200 y trae su contenido.
- Una página de producto responde 200 y contiene el botón de agregar al carrito.
- Una ruta que no existe responde 404 (detecta un servidor que contesta 200 a todo).

En local, contra un servidor de prueba en 127.0.0.1: versión nueva sana, código 0; producto con 500, código 1; versión vieja todavía sirviendo, código 1; nada escuchando, código 1.

AL TERMINAR

Checklist después del deploy

- [ ] `/healthz` devuelve la versión recién desplegada.
- [ ] El script de verificación sale con código 0.
- [ ] Recorriste a mano producto, carrito y checkout, o corriste el smoke test.
- [ ] Los registros de los primeros minutos no muestran errores nuevos.
- [ ] Si se cumplió el criterio del plan, volviste primero y estás investigando después.
- [ ] La migración de este deploy solo expande; contraer va en uno posterior, pasada la ventana.
- [ ] Anotaste qué pasó y qué cambias en el plan.