

Verificación Final Dockerfile

Verificaciones Completas

1. Stage deps ✓

- Usa `npm ci --legacy-peer-deps` (resuelve conflictos peer deps)
- Copia package.json y package-lock.json
- Verifica que node_modules se creó correctamente
- Detecta errores y muestra log completo

2. Stage builder ✓

- Copia node_modules desde stage deps
- NO copia package.json redundante (corregido #13)
- Genera Prisma client ANTES de build
- Valida enums de Prisma
- Ejecuta `npm run build` (usa npm, no yarn)
- Verifica BUILD_ID existe

3. Stage runner ✓

- Copia package.json desde builder (necesario para npm start)
- Copia node_modules completo
- Copia .next build output
- Copia scripts y shell scripts
- Ejecuta start.sh que usa `npm start`

Comandos npm/yarn verificados:

-  `npm ci --legacy-peer-deps` (deps stage)
-  `npm run build` (builder stage)
-  `npm start` (start.sh en runner)
-  `npx tsc --noEmit` (fallback para errores)
-  `npx prisma` (fallback en start.sh)

Archivos verificados en GitHub:

-  app/package.json
-  app/package-lock.json
-  app/next.config.js
-  app/tsconfig.json
-  app/prisma/schema.prisma
-  seed-admin.sh
-  backup-manual.sh
-  restore-backup.sh
-  start.sh

Errores resueltos hasta ahora:

- 1-7: Alpine, Prisma, yarn.lock, permisos, PATH
- 8: yarn.lock symlink roto
- 9-10: yarn berry config files
- 11: Yarn 4 vs Yarn 1 incompatibilidad → cambio a npm
- 12: Peer dependencies ESLint → --legacy-peer-deps
- 13: package.json copia redundante

Estado: LISTO PARA DEPLOY

El Dockerfile está optimizado y todos los errores conocidos están corregidos.