



Pre-Deploy Verification Map

Mapa completo de verificaciones del script y errores que previene.

Tabla de Mapeo: Error → Verificación

#	Error en Build	Verificación en Script	Auto-Fix
1	"/app/yarn.lock": not found	Check #1: Detecta symlink	Sí
2	Alpine 3.21 openssl-dev not found	Check #3: Valida Alpine 3.19	Manual
3	Module has no exported member 'UserRole'	Check #2: Valida 5 enums + Check #3: prisma generate	Manual
4	npm dependency conflict	Check #3: Valida yarn + --frozen-lockfile	Manual
5	Permission denied: ./start.sh	Check #6: Corrige permisos	Sí
6	Cannot find module '@prisma/client'	Check #3: Path directo prisma CLI	Manual

Verificaciones Detalladas

Check #1: yarn.lock

```

# Detecta
[ -L "yarn.lock" ]

# Corrige automáticamente
rm yarn.lock
touch yarn.lock
yarn install

# Valida
[ -f "yarn.lock" ] && [ ! -L "yarn.lock" ]

```

Previene:

- ERROR: failed to calculate checksum: "/app/yarn.lock": not found

✓ Check #2: Prisma Schema

```
# Verifica 5 enums requeridos
grep "enum UserRole" prisma/schema.prisma
grep "enum StatusCuenta" prisma/schema.prisma
grep "enum Periodicidad" prisma/schema.prisma
grep "enum TipoPago" prisma/schema.prisma
grep "enum MotivoMotarario" prisma/schema.prisma
```

Previene:

- ✗ error TS2305: Module “@prisma/client” has no exported member ‘UserRole’
 - ✗ error TS2305: Module “@prisma/client” has no exported member ‘StatusCuenta’
-

✓ Check #3: Dockerfile (6 sub-verificaciones)

3.1 Alpine Version

```
grep "alpine3.19" Dockerfile
```

Previene:

- ✗ ERROR: unable to select packages: openssl-dev (no such package)

3.2 Yarn Install

```
grep "yarn install" Dockerfile
```

Previene:

- ✗ npm ERR! Fix the upstream dependency conflict

3.3 Frozen Lockfile

```
grep "yarn install --frozen-lockfile" Dockerfile
```

Previene:

- ⚠ Versiones inconsistentes de dependencias

3.4 Prisma Generate

```
grep "prisma generate" Dockerfile
```

Previene:

- ✗ Module “@prisma/client” has no exported member ‘UserRole’

3.5 Enum Validation

```
grep "node -e.*UserRole.*require" Dockerfile
```

Previene:

- ⚠ Build completa pero enums no funcionan en runtime

3.6 Direct Prisma Path

```
grep "./node_modules/.bin/prisma generate" Dockerfile
```

Previene:

- ✗ Error: Cannot find module '@prisma/client'
-

✓ Check #4: Archivos Esenciales

```
[ -f "app/package.json" ]
[ -f "app/next.config.js" ]
[ -f "app/tsconfig.json" ]
[ -f "start.sh" ]
```

Previene:

- ✗ Cannot find module 'next/config'
 - ✗ Error loading tsconfig.json
-

✓ Check #5: Variables de Entorno

```
grep "^DATABASE_URL=" app/.env
grep "^NEXTAUTH_URL=" app/.env
grep "^NEXTAUTH_SECRET=" app/.env
```

Previene:

- ! Runtime errors por variables faltantes (informativo, Coolify las define)
-

✓ Check #6: Permisos de Scripts

```
[ -x "start.sh" ] || chmod +x start.sh
[ -x "seed-admin.sh" ] || chmod +x seed-admin.sh
[ -x "backup-manual.sh" ] || chmod +x backup-manual.sh
[ -x "restore-backup.sh" ] || chmod +x restore-backup.sh
```

Previene:

- ✗ /bin/sh: ./start.sh: Permission denied
-

Historial de Errores Resueltos

Commit Timeline

```
c984e27 - Fix: Alpine 3.21 ↗ 3.19
└> Check #3.1: Valida Alpine 3.19

7fa783b - Fix: Mejorar generación Prisma Client
└> Check #3.4: Valida prisma generate

e5d42c2 - Test: Verificación de enums
└> Check #3.5: Valida test con node -e

1408449 - Fix: npm ↗ yarn en Dockerfile
└> Check #3.2 y #3.3: Valida yarn + frozen-lockfile

df36a47 - Fix: yarn.lock symlink ↗ archivo real
└> Check #1: Detecta y corrige symlink
```

Cobertura de Verificación

Auto-Fix (2/6)

- Check #1: yarn.lock symlink
- Check #6: Permisos de scripts

Manual Fix (4/6)

- ⚠ Check #3.1: Alpine version (requiere editar Dockerfile)
- ⚠ Check #3.2-3.6: Dockerfile config (requiere editar Dockerfile)
- ⚠ Check #2: Prisma schema (requiere editar schema.prisma)
- ⚠ Check #4: Archivos esenciales (requiere crear archivos)

Uso Recomendado

Antes de Cada Deploy

```
bash pre-deploy-check.sh
```

Si encuentra errores manuales

```
# Ejemplo: Alpine 3.21 detectado
# 1. Editar Dockerfile
vim Dockerfile # Cambiar alpine3.21 → alpine3.19

# 2. Commit y re-check
git add Dockerfile
git commit -m "Fix: Use Alpine 3.19"
bash pre-deploy-check.sh
```

Workflow Completo

```
# 1. Pre-deploy check  
bash pre-deploy-check.sh  
  
# 2. Si exitoso (exit code 0)  
git add -A  
git commit -m "Ready for deploy"  
git push origin main  
  
# 3. Deploy en Coolify  
# Panel → Deploy
```

Estadísticas

- **Total de verificaciones:** 17
- **Auto-fixes:** 2
- **Warnings:** 4
- **Errores críticos detectados:** 6
- **Build failures prevenidos:** 100% (si se siguen las recomendaciones)

Última actualización: 2025-11-17

Versión del script: 2.0.0

Cobertura: 6/6 errores históricos