

# Pre-Deploy Checklist

Script automatizado que verifica y corrige problemas comunes antes de hacer deploy a Coolify.

## Uso

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

## Verificaciones que Realiza

### 1. ✓ yarn.lock

- Detecta si es un symlink (causa errores en Docker)
- Lo convierte automáticamente a archivo real
- Regenera con `yarn install` si es necesario

### 2. ✓ Prisma Schema

- Verifica que exista `prisma/schema.prisma`
- Valida los 5 enums requeridos:
  - UserRole
  - StatusCuenta
  - Periodicidad
  - TipoPago
  - MotivoMotarario

### 3. ✓ Dockerfile

- **Alpine 3.19:** Verifica versión correcta (no 3.21 que causa errores)
- **yarn + frozen-lockfile:** Confirma uso de `yarn install --frozen-lockfile`
- **Prisma Generate:** Valida que ejecute `prisma generate`
- **Validación de Enums:** Verifica test con `node -e` para UserRole
- **Path Directo:** Confirma uso de `./node_modules/.bin/prisma` (no npx)

### 4. ✓ Archivos Esenciales

- `app/package.json`
- `app/next.config.js`
- `app/tsconfig.json`
- `start.sh`

### 5. ✓ Variables de Entorno

- DATABASE\_URL
- NEXTAUTH\_URL
- NEXTAUTH\_SECRET

## 6. ✓ Permisos de Scripts

Corrige automáticamente permisos de:

- start.sh
- seed-admin.sh
- backup-manual.sh
- restore-backup.sh

## Exit Codes

- **0**: Todo OK, listo para deploy
- **1**: Se encontraron problemas, revisar output

## Workflow Recomendado

```
# 1. Ejecutar pre-deploy check
bash pre-deploy-check.sh

# 2. Si todo OK, hacer commit y push
git add -A
git commit -m "Ready for deploy"
git push origin main

# 3. Deploy en Coolify
# Ve al panel de Coolify y haz clic en "Deploy"
```

## Problemas Comunes que Resuelve

### ✗ Error 1: yarn.lock symlink roto

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

**Causa:** yarn.lock era un symlink a /opt/hostedapp/node/root/app/yarn.lock

**Solución:** Script detecta y convierte automáticamente a archivo real

**Verificación:** Check #1 - Convierte symlink → archivo real (448KB)

### ✗ Error 2: Alpine Linux 3.21 Repository Error

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

**Causa:** Alpine 3.21 tiene problemas con repositorios

**Solución:** Script verifica que Dockerfile use Alpine 3.19

**Verificación:** Check #3 - Valida FROM node:18-alpine3.19

## ✗ Error 3: Prisma Client - Enums no exportados

```
error TS2305: Module '"@prisma/client"' has no exported member 'UserRole'  
error TS2305: Module '"@prisma/client"' has no exported member 'StatusCuenta'
```

**Causa:** Prisma Client no se generó correctamente o enums no disponibles

**Solución:** Script verifica:

- Prisma schema tiene los 5 enums requeridos (Check #2)
- Dockerfile ejecuta `prisma generate` (Check #3)
- Dockerfile valida enums con `node -e test` (Check #3)

**Verificación:** Check #2 y #3 - Schema + Generación + Validación

---

## ✗ Error 4: npm vs yarn inconsistencia

```
npm ERR! Fix the upstream dependency conflict, or retry with --legacy-peer-deps
```

**Causa:** Proyecto usa yarn.lock pero Dockerfile usaba npm

**Solución:** Script verifica que Dockerfile use yarn + `--frozen-lockfile`

**Verificación:** Check #3 - Valida `yarn install --frozen-lockfile`

---

## ✗ Error 5: Scripts sin permisos

```
/bin/sh: ./start.sh: Permission denied
```

**Causa:** Scripts no tienen permisos de ejecución

**Solución:** Script corrige permisos automáticamente

**Verificación:** Check #6 - Ejecuta `chmod +x` automáticamente

---

## ✗ Error 6: npx prisma genera problemas

```
Error: Cannot find module '@prisma/client'
```

**Causa:** `npx prisma` puede no encontrar el CLI correcto

**Solución:** Script verifica uso de `./node_modules/.bin/prisma`

**Verificación:** Check #3 - Path directo a Prisma CLI

---

## ✗ Error 7: Reinstalación de yarn en Alpine

```
npm error EEXIST: file already exists  
npm error File exists: /usr/local/bin/yarn
```

**Causa:** RUN `npm install -g yarn` en Dockerfile cuando yarn ya viene preinstalado

**Solución:** Script detecta y rechaza instalación redundante

**Verificación:** Check #3 - No reinstalar yarn en Alpine

## Integración con CI/CD

Puedes agregar este script como paso de pre-deploy en tu pipeline:

```
# .github/workflows/deploy.yml
- name: Pre-deploy check
  run: bash pre-deploy-check.sh
```

## Output Ejemplo

```
🔍 PRE-DEPLOY CHECK
=====

i 1. Verificando yarn.lock...
✅ yarn.lock es un archivo válido (448K)

i 2. Verificando Prisma schema...
✅ Prisma schema válido (5/5 enums encontrados)

i 3. Verificando Dockerfile...
✅ Dockerfile usa Alpine 3.19 (correcto)
✅ Dockerfile usa yarn (correcto para yarn.lock)
✅ Dockerfile usa --frozen-lockfile (correcto)
✅ Dockerfile genera Prisma client
✅ Dockerfile valida enums de Prisma
✅ Dockerfile usa path directo a prisma CLI (correcto)

i 4. Verificando archivos esenciales...
✅ app/package.json ✅
✅ app/next.config.js ✅
✅ app/tsconfig.json ✅
✅ start.sh ✅

i 5. Verificando .env (opcional)...
✅ DATABASE_URL definido
✅ NEXTAUTH_URL definido
✅ NEXTAUTH_SECRET definido

i 6. Verificando permisos de scripts...
✅ start.sh tiene permisos de ejecución
✅ seed-admin.sh tiene permisos de ejecución
✅ backup-manual.sh tiene permisos de ejecución
✅ restore-backup.sh tiene permisos de ejecución

=====
✅ 🎉 i TODO LISTO PARA DEPLOY!

i Próximos pasos:
1. git add -A
2. git commit -m 'Pre-deploy check: Todo OK'
3. git push origin main
4. Deploy en Coolify
```

## Notas

---

- El script es **idempotente**: puedes ejecutarlo múltiples veces sin problemas
  - **Auto-corrige** problemas cuando es posible
  - **No requiere** argumentos ni configuración
  - **Compatible** con cualquier entorno Linux/Unix
- 

**Última actualización:** 2025-11-17

**Versión:** 1.0.0