



PREVENCIÓN: Problema yarn.lock Symlink



El Problema

Error en Docker Build:

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

Causa: El archivo `yarn.lock` es un **symlink** (enlace simbólico) que Docker no puede copiar durante el build.

Impacto: El deploy falla en EasyPanel, Coolify, Railway, Render, y cualquier plataforma que use Docker.



SOLUCIÓN PREVENTIVA IMPLEMENTADA

He creado 4 scripts automáticos para prevenir este problema:

1. `scripts/fix-yarn-lock-symlink.sh`

Propósito: Convertir yarn.lock de symlink a archivo real

Uso:

```
./scripts/fix-yarn-lock-symlink.sh
```

Qué hace:

- ☒ Detecta si yarn.lock es un symlink
- ☒ Lo convierte a archivo regular
- ☒ Verifica que la conversión fue exitosa

2. `scripts/pre-push-check.sh`

Propósito: Verificar yarn.lock antes de cada push

Uso:

```
./scripts/pre-push-check.sh
```

Qué verifica:

- ☒ Si yarn.lock existe
- ☒ Si es un symlink (y lo convierte si es necesario)
- ☒ Si el tamaño es válido (>10KB)
- ☒ Previene push si hay problemas

3. `scripts/setup-git-hooks.sh`

Propósito: Instalar git hook automático

Uso (una sola vez):

```
./scripts/setup-git-hooks.sh
```

Qué hace:

- ☒ Instala un pre-push hook en `.git/hooks/`
- ☒ Hace ejecutables todos los scripts
- ☒ Ejecuta automáticamente verificaciones antes de cada push

IMPORTANTE: Ejecuta este script UNA VEZ al inicio para activar las verificaciones automáticas.

4. `scripts/safe-push.sh`

Propósito: Push seguro con todas las verificaciones

Uso:

```
./scripts/safe-push.sh
```

Qué hace:

1. ☒ Verifica estado de git
2. ☒ Ofrece commitear cambios pendientes
3. ☒ Verifica yarn.lock automáticamente
4. ☒ Hace pull antes de push (evita conflictos)
5. ☒ Hace push seguro
6. ☒ Muestra el último commit

Uso con token:

```
GITHUB_TOKEN=tu_token ./scripts/safe-push.sh
```

**FLUJO DE TRABAJO RECOMENDADO****Opción A: Uso Manual (Recomendado para Principiantes)**

```
# 1. Hacer cambios en el código
# 2. Verificar yarn.lock antes de push
./scripts/pre-push-check.sh

# 3. Si todo está OK, hacer push normal
git add -A
git commit -m "feat: Nueva funcionalidad"
git push origin main
```

Opción B: Uso Automático (Recomendado)

```
# 1. Instalar git hooks (una sola vez)
./scripts/setup-git-hooks.sh

# 2. Hacer cambios en el código
# 3. Push normal - las verificaciones se ejecutan automáticamente
git add -A
git commit -m "feat: Nueva funcionalidad"
git push origin main
```

Opción C: Push Todo-en-Uno (Más Seguro)

```
# 1. Hacer cambios en el código
# 2. Ejecutar script seguro
./scripts/safe-push.sh
```

Este script hace TODO automáticamente:

- ☒ Verifica cambios
- ☒ Commitea si es necesario
- ☒ Verifica yarn.lock
- ☒ Hace pull para evitar conflictos
- ☒ Hace push



CASOS DE USO

Caso 1: Convertir yarn.lock Manualmente

Si sabes que yarn.lock es un symlink:

```
./scripts/fix-yarn-lock-symlink.sh
git add app/yarn.lock
git commit -m "fix: Convertir yarn.lock a archivo regular"
git push origin main
```

Caso 2: Verificar Antes de Push

Antes de hacer push, siempre puedes verificar:

```
./scripts/pre-push-check.sh
```

Si todo está OK, procede con push normal.

Caso 3: Push Completamente Seguro

Si quieres máxima seguridad:

```
./scripts/safe-push.sh
```

Caso 4: Push con Token de GitHub

Si usas Personal Access Token:

```
GITHUB_TOKEN=ghp_tu_token_aqui ./scripts/safe-push.sh
```

O configura la variable permanentemente:

```
export GITHUB_TOKEN=ghp_tu_token_aqui
./scripts/safe-push.sh
```



ENTENDIENDO EL PROBLEMA

¿Por Qué Ocurre?

El sistema de paquetes de Yarn a veces crea `yarn.lock` como symlink por optimización de espacio o cuando se usa Yarn PnP (Plug'n'Play).

Estructura típica:

```
app/yarn.lock -> /opt/hostedapp/node/root/app/yarn.lock
```

¿Por Qué Docker No Puede Copiarlo?

Durante el build de Docker, el `COPY` solo ve el symlink, no el archivo destino. El contexto de Docker no incluye `/opt/hostedapp/`, por lo que falla.

¿Cómo Prevenirlo?

1. **Siempre usar estos scripts** antes de push
2. **Instalar git hooks** (recomendado)
3. **Evitar comandos Yarn PnP** si es posible
4. **Usar `yarn install normal`**, no `yarn pnp install`



CHECKLIST DE INSTALACIÓN

Primera Vez (Setup Inicial)

- [] Ejecutar `./scripts/setup-git-hooks.sh` una vez
- [] Verificar que scripts son ejecutables (`chmod +x scripts/*.sh`)
- [] Probar con `./scripts/pre-push-check.sh`
- [] Hacer un push de prueba (debería verificar automáticamente)

Antes de Cada Deploy

- [] Verificar que `yarn.lock` no es symlink
- [] Ejecutar `./scripts/pre-push-check.sh` si tienes dudas
- [] O usar `./scripts/safe-push.sh` para máxima seguridad

TROUBLESHOOTING

Problema: “Permission denied” al ejecutar scripts

Solución:

```
chmod +x scripts/*.sh
```

Problema: Git hook no se ejecuta automáticamente

Solución:

```
./scripts/setup-git-hooks.sh
```

Esto reinstalará el hook.

Problema: yarn.lock sigue siendo symlink después de push

Causa: El git hook no está instalado o no se ejecutó.

Solución:

```
# 1. Instalar hooks
./scripts/setup-git-hooks.sh

# 2. Convertir yarn.lock
./scripts/fix-yarn-lock-symlink.sh

# 3. Commitear y push
git add app/yarn.lock
git commit -m "fix: Convertir yarn.lock a archivo regular"
git push origin main
```

Problema: Docker build sigue fallando con “yarn.lock not found”

Posibles causas:

1. El último push aún tenía yarn.lock como symlink
2. Cache de Docker en EasyPanel/Coolify

Solución:

```
# 1. Verificar en local
ls -la app/yarn.lock
# Debe mostrar: -rw-r--r-- (archivo regular)
# NO debe mostrar: lrwxrwxrwx (symlink)

# 2. Si es symlink, convertir:
./scripts/fix-yarn-lock-symlink.sh
git add app/yarn.lock
git commit -m "fix: Convertir yarn.lock a archivo regular"
git push origin main

# 3. En EasyPanel/Coolify:
# - Force Rebuild
# - Clear cache si está disponible
```

ESTADÍSTICAS

Commits afectados por este problema:

- 2776a27 - Primera vez que se resolvió
- c84e37a - Segunda vez que reapareció

Tiempo perdido: ~30-60 minutos cada vez

Tiempo ahorrado con prevención: ∞ (no volverá a ocurrir)

RECOMENDACIÓN FINAL

Para máxima prevención:

1. Instala los git hooks una vez:

```
bash
./scripts/setup-git-hooks.sh
```

2. Usa safe-push para todo:

```
bash
# En lugar de git push origin main
./scripts/safe-push.sh
```

3. Verifica después del deploy:

- Si el build falla con "yarn.lock not found"
 - Ejecuta inmediatamente: `./scripts/fix-yarn-lock-symlink.sh`
-

DOCUMENTOS RELACIONADOS

- `FIX_YARN_LOCK_SYMLINK.md` - Análisis del problema original
 - `ESTADO_FINAL_DEPLOY.md` - Guía completa de deploy
 - `CONFIGURACION_EASYPANEL_CORRECTA.md` - Configuración de EasyPanel
-

Última actualización: 25 de Octubre, 2025

Autor: Sistema de Prevención Automática

Estado:  Activo y funcionando