

Fix: Symlink node_modules.backup causando error ENOENT en build

Fecha: 28 de octubre de 2025

Commit: Por aplicar

Tipo: Corrección crítica de build

Problema Identificado

El build de Docker fallaba durante `npm run build` con:

```
[Error: ENOENT: no such file or directory, stat '/app/node_modules.backup'] {
  errno: -2,
  code: 'ENOENT',
  syscall: 'stat',
  path: '/app/node_modules.backup'
}
```

Causa Raíz

Un **symlink problemático** existía en `app/node_modules.backup` :

```
app/node_modules.backup -> /opt/hostedapp/node/root/app/node_modules
```

Este symlink:



1. Apuntaba a un path absoluto específico del host de desarrollo
2. Se copiaba al contenedor Docker durante `COPY app/ ./`
3. Next.js intentaba seguir el symlink durante el build
4. El path no existía en el contenedor → Error ENOENT

Soluciones Aplicadas

1. Eliminar el symlink problemático

```
rm -f app/node_modules.backup
```

Resultado:

-  Symlink eliminado del filesystem
-  No hay más referencias a paths absolutos del host

2. Actualizar `.dockerignore`

Agregadas reglas más robustas para prevenir que se copien backups/symlinks en el futuro:

```
# Backups
*.backup
*.bak
*_BACKUP_*
node_modules.backup      # ← Nuevo: específico para este caso
**/*.backup              # ← Nuevo: en cualquier subdirectorio
**/*_BACKUP_*            # ← Nuevo: patrón más robusto
```

3. Validación post-fix

```
# Verificar que no hay symlinks problemáticos
find app/ -maxdepth 2 -type l -name "*.backup"
# Resultado: ✅ Ninguno encontrado

# Verificar que no hay referencias en el código
grep -r "node_modules.backup" app/*.{js,ts,json}
# Resultado: ✅ Ninguna referencia encontrada
```

Resultado Esperado

Ahora el build de Docker debe completar sin errores:

1. ✅ COPY app/ ./ no copia symlinks .backup
2. ✅ Next.js no intenta acceder a node_modules.backup
3. ✅ Build completa exitosamente
4. ✅ Imagen Docker se genera correctamente

Archivos Modificados

```
app/node_modules.backup - ❌ Eliminado (era un symlink)
.dockerignore           - ✅ Actualizado con reglas más robustas
```

Próximos Pasos

1. Commit y push de cambios
2. Rebuild en EasyPanel (limpiar caché)
3. Verificar que el build completa sin error ENOENT
4. Confirmar que la app inicia correctamente

Prevención Futura

Buenas prácticas para evitar este tipo de problemas:

1. **No crear symlinks con paths absolutos** en el directorio del proyecto
2. **Revisar .dockerignore regularmente** para excluir archivos temporales
3. **Validar el contexto de build** antes de hacer push:

```
bash
# Ver qué se copiará al contenedor
docker build --no-cache --target builder -t test . --progress=plain
```

4. **Usar .gitignore efectivo** para que estos archivos no lleguen al repo:

```
gitignore
node_modules.backup
*.backup
```

Validación Pre-Deploy

Checklist verificado:

- [x] Symlink node_modules.backup eliminado
- [x] .dockerignore actualizado
- [x] No hay symlinks problemáticos en app/
- [x] No hay referencias a node_modules.backup en el código
- [x] package.json tiene todos los scripts
- [x] prisma está en devDependencies
- [x] Estructura del proyecto correcta

Contexto Histórico

Este symlink probablemente fue creado durante:

- Testing local con diferentes versiones de node_modules
- O por un proceso de backup automático

Lección: Los symlinks relativos son seguros, pero los absolutos causan problemas en Docker y CI/CD.

Estado:  Listo para deploy

Prioridad:  CRÍTICA

Testing: Validación local completada