


# Resumen Completo de Fixes - 30 de Octubre de 2025

---


## Fixes Aplicados Hoy

---


### 1. Error Dockerfile: Redirección en COPY (Commit 81ed919)

- **Problema:** `COPY app/.yarn* ./ 2>/dev/null || true` causaba error
- **Solución:** Eliminada línea - archivos `.yarn*` no son críticos
- **Estado:**  Resuelto


### 2. yarn.lock como symlink (Commit f55dd31)

- **Problema:** Docker no puede copiar symlinks
- **Solución:** Convertido a archivo regular (495KB)
- **Estado:**  Resuelto


### 3. node\_modules not found en stage builder (Commit 150337c)

- **Problema:** `COPY --from=deps /app/node_modules: not found`
- **Solución:** Agregadas verificaciones explícitas en stage deps
- **Estado:**  Resuelto

### 4. Sin verificaciones pre-build (Commit 150337c)


- **Problema:** No se detectaban problemas antes de push/build
- **Solución:** Creado script `pre-build-check.sh` (24 verificaciones)
- **Estado:**  Resuelto

### 5. Dockerfile usando package-lock.json (Fixes anteriores)

- **Problema:** Proyecto usa Yarn, no NPM
- **Solución:** Actualizado Dockerfile para usar solo Yarn
- **Estado:**  Resuelto

## Estado Actual del Proyecto

---

Repositorio: <https://github.com/qhosting/escalafin> (main)  
Mirror: <https://github.com/qhosting/escalafinmx> (main)  
ltimo commit: 150337c  
Versión: 1.1.1  
Build: 20251030.004 (pendiente actualización)

## Archivos Críticos Actualizados

Archivo	Estado	Descripción
Dockerfile	✓ Actualizado	Verificaciones explícitas de node_modules
app/yarn.lock	✓ Archivo regular	No symlink (495KB)
app/package.json	✓ OK	Dependencias Yarn
.dockerignore	✓ OK	Incluye scripts production
scripts/pre-build-check.sh	✓ NUEVO	24 verificaciones completas
scripts/pre-push-check.sh	✓ Actualizado	Verifica archivos críticos
scripts/push-ambos-repos.sh	✓ OK	Verifica yarn.lock
scripts/fix-yarn-lock-symlink.sh	✓ OK	Auto-convierte symlinks

## Instrucciones para Deploy en EasyPanel

### Paso 1: Pull del Último Commit

```
cd /ruta/a/escalafin
git pull origin main
```

Verificar que esté en commit 150337c :

```
git log -1 --oneline
# Debería mostrar: 150337c fix: agregar verificaciones explícitas de node_modules
```

### Paso 2: Clear Build Cache

En el panel de EasyPanel:

1. Ir a la aplicación EscalaFin
2. Click en **“Rebuild”**
3. Seleccionar **“Clear cache and rebuild”**
4. Confirmar

### Paso 3: Monitorear Build

Observar logs en tiempo real. **Ahora verás mensajes claros:**

**Stage DEPS (nuevo):**

```

📦 Instalando dependencias con Yarn...
✅ Yarn install completado

🔍 Verificando node_modules...
✅ node_modules generado: 450 paquetes instalados
✅ Dependencias instaladas correctamente

```

#### Lo que NO debe aparecer:

- ❌ `lstat /2>/dev/null` (ya resuelto)
- ❌ `node_modules` no fue generado (indica yarn install falló)
- ❌ `node_modules` parece vacío (indica instalación parcial)
- ❌ `COPY --from=deps /app/node_modules: not found` (ya resuelto)

#### Stage BUILDER:

- Confirmar que `yarn prisma generate` funciona
- Verificar que Next.js build termina sin errores

### Paso 4: Verificar Scripts en Container

Una vez que el container esté corriendo:

```

docker exec -it escalafin ls -lah /app/

# Debería incluir:
# -rwxr-xr-x start-improved.sh
# -rwxr-xr-x emergency-start.sh
# -rwxr-xr-x healthcheck.sh

```

### Paso 5: Verificar Logs de Startup

```

docker logs escalafin -f

# Debería mostrar:
# ✅ Node version: v18.x.x
# ✅ Yarn version: 4.10.3
# 🔄 Ejecutando migraciones de Prisma...
# ✅ Migraciones completadas
# 🚀 Iniciando servidor Next.js...
# ✓ Ready in X.XXs

```

### Paso 6: Health Check

```

curl https://escalafin.com/api/health

# Respuesta esperada:
{
  "status": "ok",
  "timestamp": "2025-10-30T...",
  "version": "1.1.1"
}

```

## Paso 7: Verificar Versión

```
curl https://escalafin.com/api/system/version
```

```
# Respuesta esperada:
{
  "version": "1.1.1",
  "build": "20251030.003",
  "commit": "f55dd31",
  "environment": "production"
}
```

### Troubleshooting

#### Si el build falla en “COPY app/.yarn\*”:

**Ya está resuelto** - ese comando fue eliminado del Dockerfile.

#### Si aparece “yarn.lock is a symlink”:

**Ya está resuelto** - yarn.lock es ahora un archivo regular.

#### Si persisten errores de Prisma:

```
# En el container:
docker exec -it escalafin yarn prisma generate
docker exec -it escalafin yarn prisma migrate deploy
```

#### Si Next.js no inicia:

```
# Usar el script de emergencia:
docker exec -it escalafin /app/emergency-start.sh
```



## Comandos de Verificación Rápida

```
# 1. Estado del repo
cd /home/ubuntu/escalafin_mvp
git status
git log -3 --oneline

# 2. Verificar yarn.lock
ls -lah app/yarn.lock
# Debe ser un archivo (-rw-r--r--), NO un symlink (lrwxrwxrwx)

# 3. Test local del Dockerfile (opcional)
cd /home/ubuntu/escalafin_mvp
docker build -t escalafin-test:local .

# 4. Verificar scripts de producción
ls -lah app/*.sh
# start-improved.sh
# emergency-start.sh
# healthcheck.sh
```

## ✓ Checklist Pre-Deploy

---

- [x] Dockerfile corregido (sin redirecciones en COPY)
- [x] yarn.lock convertido a archivo regular
- [x] Scripts de producción presentes (.dockerignore actualizado)
- [x] Pusheado a ambos repos (escalafin + escalafinmx)
- [x] Documentación completa generada
- [x] Sistema de versionado implementado
- [x] Pre-push hooks configurados

## 🎯 Siguiete Acción Inmediata

---

### EN EASYPANEL:

1. Pull del commit `f55dd31`
2. Clear cache + Rebuild
3. Verificar logs de build
4. Confirmar que la app inicia correctamente
5. Validar health check y versión

## 📖 Documentación Relacionada

---

- `FIX_NODE_MODULES_VERIFICATION_30_OCT_2025.md` - ★ Fix verificación node\_modules (NUEVO)
  - `FIX_DOCKERFILE_COPY_ERROR_30_OCT_2025.md` - Fix del error COPY
  - `FIX_DOCKERFILE_YARN_30_OCT_2025.md` - Cambios de NPM a Yarn
  - `MIGRACION_ESCALAFINMX_29_OCT_2025.md` - Setup dual repos
  - `SISTEMA_VERSIONADO.md` - Sistema de versiones
- 

**Última actualización:** 30 de octubre de 2025, 02:35 AM

**Commit actual:** 150337c

**Estado:** ✓ Listo para deploy con verificaciones completas