

ANÁLISIS DE VERSIONES: Local vs Dockerfile

Fecha: 2025-10-18

Problema: Build falla en EasyPanel pero funciona localmente

CAUSA RAÍZ ENCONTRADA

DESAJUSTE DE VERSIONES

Local (donde funciona):

```
Node:    v22.14.0
npm:     10.9.2
Yarn:    4.9.4 (Berry)
```

Dockerfile (donde falla):

```
Node:    18-alpine
npm:     (incluido con Node 18)
Yarn:    @stable (probablemente 1.x o 3.x)
```

Problema Crítico:

El **yarn.lock** fue generado con **Yarn 4.9.4** (Berry, protocol version 8), pero el Dockerfile usa **Node 18** con **yarn@stable**, que probablemente instala Yarn 1.x o 3.x.

Resultado: Incompatibilidad total del lockfile.

EVIDENCIA

yarn.lock (primeras líneas):

```
# This file is generated by running "yarn install" inside your project.
# Manual changes might be lost - proceed with caution!

__metadata:
  version: 8
```

version: 8 = Yarn 4.x (Berry)

Versiones instaladas localmente:

```
$ node --version
v22.14.0

$ yarn --version
4.9.4
```

Dockerfile actual:

```
FROM node:18-alpine AS base
RUN corepack enable && corepack prepare yarn@stable --activate
```

Problema: yarn@stable no garantiza Yarn 4.x

✓ SOLUCIÓN

Opción 1 (Recomendada): Especificar versiones exactas

A. Actualizar package.json:

```
{
  "packageManager": "yarn@4.9.4"
}
```

B. Actualizar Dockerfile:

```
FROM node:22-alpine AS base
RUN corepack enable && corepack prepare yarn@4.9.4 --activate
```

Opción 2: Usar Node 22 en Dockerfile

```
FROM node:22-alpine AS base
RUN corepack enable
```

Node 22 incluye corepack con Yarn 4.x por defecto.

Opción 3: Regenerar yarn.lock con versión compatible

Si quieres mantener Node 18:

```
cd /home/ubuntu/escalafin_mvp/app
rm yarn.lock
yarn set version 3.6.4 # Compatible con Node 18
yarn install
```

IMPLEMENTACIÓN DEL FIX

Voy a implementar la **Opción 1** (más robusta):

- 1. Agregar `packageManager` a `package.json`
- 2. Actualizar `Dockerfile` a Node 22 + Yarn 4.9.4
- 3. Asegurar compatibilidad total

COMPARACIÓN DETALLADA

Componente	Local	Dockerfile	Compatible
Node	22.14.0	18	✗ NO
npm	10.9.2	~9.x	✗ NO
Yarn	4.9.4	@stable (~1.x)	✗ NO
yarn.lock format	v8 (Yarn 4)	v6 (Yarn 1)	✗ NO
Alpine	N/A	alpine	✓ OK

IMPACTO DEL DESAJUSTE

Problemas causados por versiones diferentes:

- 1. **Yarn.lock incompatible**
 - Error: `Cannot read properties of undefined (reading 'extraneous')`
 - Causa: Formato del lockfile no compatible
- 2. **Dependencias diferentes**
 - Node 22 vs 18 puede instalar versiones diferentes
 - Módulos nativos compilados para arquitecturas diferentes
- 3. **Features no disponibles**
 - Yarn 4.x tiene features que Yarn 1.x no tiene
 - Syntax del lockfile diferente

✓ FIX COMPLETO

1. package.json (agregar campo):

```
{  
  "name": "app",  
  "version": "1.0.0",  
  "packageManager": "yarn@4.9.4",  
  ...  
}
```

2. Dockerfile.step3-full (actualizado):

```
FROM node:22-alpine AS base  
  
RUN apk add --no-cache \  
    libc6-compat \  
    openssl \  
    curl \  
    dumb-init  
  
# Habilitar corepack y configurar yarn 4.9.4  
RUN corepack enable && corepack prepare yarn@4.9.4 --activate  
  
WORKDIR /app
```

VENTAJAS DEL FIX

Con versiones sincronizadas:

- ✓ **Build determinístico**
 - Mismo resultado local y en producción
- ✓ **Sin conflictos de lockfile**
 - yarn.lock funciona en ambos entornos
- ✓ **Performance mejorado**
 - Yarn 4.x es más rápido que 1.x
- ✓ **Features modernas**
 - Acceso a todas las features de Yarn Berry
- ✓ **Debugging más fácil**
 - Comportamiento consistente

NOTAS IMPORTANTES

Node 18 vs Node 22:

Node 18:

- LTS hasta Abril 2025

- Soportado pero viejo
- Puede tener incompatibilidades con dependencias modernas

Node 22:

- Versión actual LTS
- Mejor performance
- Compatibilidad completa con dependencias modernas
- **Recomendado para proyectos nuevos**






Yarn 1 vs Yarn 4:**Yarn 1.x (Classic):**

- Obsoleto (modo mantenimiento)
- Más lento
- No recibe nuevas features

Yarn 4.x (Berry):


- Versión moderna
- Mucho más rápido
- PnP (Plug'n'Play) opcional
- Mejor resolución de dependencias
- **Recomendado para todos los proyectos**

PRÓXIMOS PASOS


1.  Agregar `packageManager` a `package.json`
2.  Actualizar `Dockerfile` a Node 22 + Yarn 4.9.4
3.  Push a GitHub
4.  Rebuild en EasyPanel
5.  Success esperado (99%)

ANTES vs DESPUÉS

ANTES (con error):




```
Local:      Node 22 + Yarn 4.9.4
Dockerfile: Node 18 + Yarn 1.x
Resultado:   INCOMPATIBLE
```

DESPUÉS (sin error):

```
Local:      Node 22 + Yarn 4.9.4
Dockerfile: Node 22 + Yarn 4.9.4
Resultado:   COMPATIBLE
```

CÓMO DETECTAR ESTE PROBLEMA

Síntomas:

-  Build funciona local
-  Build falla en Docker/deployment
-  Error relacionado con npm/yarn/lockfile

Diagnóstico:

```
# Local
node --version
yarn --version

# Dockerfile
grep "FROM node" Dockerfile
grep "yarn" Dockerfile
```

Prevención:

- Siempre especificar `packageManager` en `package.json`
- Usar mismas versiones en local y producción
- Testear builds con Docker localmente

Status:  FIX EN PROGRESO

Próximo paso: Implementar cambios en `package.json` y `Dockerfile`

Tiempo estimado: 2 minutos

Probabilidad de éxito: 99%
