

✓ FIX: Error “.next directory not found”

Fecha: 18 de octubre de 2025

Error Original:

Error: Could not find a production build **in** the **'next'** directory

🔍 PROBLEMA IDENTIFICADO

El error **NO** era en el **build**, sino en el **runtime del contenedor**.

Causa Raíz

Next.js con `outputFileTracingRoot` genera una estructura anidada:

```
.next/standalone/  
└─ app/ ── El contenido del directorio "app"  
    └─ .next/  
        └─ server.js  
        └─ package.json  
        └─ node_modules/
```

El Dockerfile estaba copiando mal:

```
# ✗ ANTES (INCORRECTO)  
COPY --from=builder /app/.next/standalone ./  
  
# Esto copiaba:  
/app/  
└─ app/ ── Carpeta extra  
    └─ .next/ ── Next.js busca en /app/.next pero está aquí
```

✓ SOLUCIÓN APLICADA

Copiar desde `.next/standalone/app/` en lugar de `.next/standalone/` :

```
# ✓ AHORA (CORRECTO)  
COPY --from=builder /app/.next/standalone/app ./  
  
# Esto copia directamente:  
/app/  
└─ .next/ ── Next.js lo encuentra aquí  
    └─ server.js ── start.sh lo encuentra aquí  
    └─ node_modules/
```

CAMBIOS REALIZADOS

1. Dockerfile - Runtime Stage

Antes:

```
COPY --from=builder /app/.next/standalone ./
```

Después:

```
# Copy standalone build (con outputFileTracingRoot, standalone contiene carpeta app/)
COPY --from=builder /app/.next/standalone/app ./
```

2. Verificaciones Añadidas

```
# Verificar estructura del standalone
RUN echo "👉 Verificando estructura del standalone..." && \
  ls -la .next/standalone/ && \
  echo "" && \
  ls -la .next/standalone/app/ && \
  test -f ".next/standalone/app/server.js" || \
    (echo "❌ Error: server.js no encontrado en standalone/app/" && exit 1)
```

RESULTADO ESPERADO

Build Phase

- ✓ yarn install
- ✓ npx prisma generate
- ✓ yarn build
- ✓ .next/standalone generado
- ✓ .next/standalone/app/server.js encontrado

Runtime Phase

- ✓ Archivos copiados a /app/
- ✓ server.js en /app/server.js
- ✓ .next/ en /app/.next/
- ✓ node server.js ejecuta correctamente
- ✓ Next.js encuentra el build

PASOS PARA APLICAR EL FIX

1 Pull Latest Commit

```
git pull origin main
```

Commit actual: [será actualizado al hacer push]

2 Limpia el Cache en EasyPanel

- Settings > Build > **Clear Build Cache**

3 Configura Recursos

Memory: 2GB
CPU: 1-2 vCPUs
Timeout: 600s

4 Rebuild

Haz clic en **Deploy/Rebuild**

QUÉ VERÁS EN LOS LOGS

Durante Build

```
🔧 Building Next.js...
✓ Compiled successfully
✓ Generating static pages (59/59)
✓ Build completado

📁 Verificando estructura del standalone...
drwxr-xr-x app
-rw-r--r-- server.js
✓ server.js encontrado en standalone/app/
```

Durante Runtime

```
🚀 Iniciando ESCALAFIN...
🔍 Verificando archivos de Next.js standalone...
✓ server.js encontrado en /app/server.js (CORRECTO)
🚀 Iniciando servidor Next.js standalone...
📁 Working directory: /app
💻 Server: /app/server.js
🌐 Hostname: 0.0.0.0
🔊 Port: 3000
🎉 EJECUTANDO: node server.js

▲ Next.js 14.2.28
- Local: http://0.0.0.0:3000
✓ Ready in XXXms
```

POSIBLES ERRORES Y SOLUCIONES

Si Falla el Build

Error:

```
standalone no generado
```

Causa: `NEXT_OUTPUT_MODE` no configurado

Solución:

En Dockerfile ya está: `ENV NEXT_OUTPUT_MODE=standalone`

Si Falla en Runtime

Error:

```
server.js no encontrado en standalone/app/
```

Causa: Next.js no generó el standalone correctamente

Solución:

Verifica que `output: process.env.NEXT_OUTPUT_MODE` esté en `next.config.js`

Si Falla al Iniciar

Error:

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

Causa: Prisma no se copió correctamente

Solución:

Verifica que estas líneas estén en el Dockerfile:

```
COPY --from=builder /app/prisma ./prisma
COPY --from=builder /app/node_modules/.prisma ./node_modules/.prisma
```

CONFIANZA

95% de que este fix resolverá el problema porque:

- ✓ El build ya funcionaba (Next.js compilaba sin errores)
- ✓ Solo faltaba copiar los archivos correctamente
- ✓ La verificación asegura que `server.js` esté en el lugar correcto
- ✓ El `start.sh` busca `server.js` en `/app/server.js`
- ✓ Todo está alineado ahora

CONTEXTO TÉCNICO

Next.js Standalone Output

Cuando Next.js genera el standalone con `outputFileTracingRoot` :






```
// next.config.js
experimental: {
  outputFileTracingRoot: path.join(__dirname, '../'),
}
```

Crea esta estructura:





```
proyecto/
├── app/
│   └── .next/
│       └── standalone/
│           └── app/
│               ├── .next/
│               ├── server.js
│               └── node_modules/
└── Replica la estructura desde root
```

Por eso necesitamos copiar desde `.next/standalone/app/` y no solo `.next/standalone/`.

HISTORIAL DE FIXES

1.  **Error de npm extraneous** → Cambio a Yarn
2.  **Yarn PnP mode** → Agregado `.yarnrc.yml`
3.  **yarn.lock symlink** → Convertido a archivo regular
4.  **Build debugging** → Agregado logging completo
5.  **Runtime error** → **Fix de la estructura del standalone** ← ESTE FIX

PRÓXIMOS PASOS

1. **Pull** el commit actualizado
2. **Limpia** el cache en EasyPanel
3. **Rebuild** y observa los logs
4. Deberías ver:
 -  Build exitoso
 -  Verificación del standalone
 -  Runtime iniciando correctamente
 -  Aplicación corriendo en puerto 3000

Última actualización: 18 de octubre de 2025

Status:  FIX APLICADO - LISTO PARA REBUILD

¡Este debería ser el último fix necesario! 🚀