

Dockerfile v8.9 - Fix Compatibilidad Shell

❌ Problema en v8.8

```
ERROR: exit code: 2
```

No es el build de Next.js, es el script de shell.

Causa Raíz

```
BUILD_EXIT_CODE=${PIPESTATUS[0]}
```

Problema:

- PIPESTATUS solo existe en **Bash**
- Alpine Linux usa `/bin/sh` (no bash)
- `/bin/sh` no reconoce PIPESTATUS
- **Exit code 2** = error de sintaxis en shell

✅ Solución en v8.9

Sintaxis Compatible con sh

```
RUN set -o pipefail && \
  echo "=== BUILD NEXT.JS ===" && \
  (npm run build 2>&1 | tee /tmp/build-output.log) || \
  (echo "❌ BUILD FALLÓ" && \
  echo "=== ÚLTIMAS 100 LÍNEAS DEL LOG ===" && \
  tail -100 /tmp/build-output.log && \
  echo "=== ARCHIVOS EN /app ===" && \
  ls -la && \
  echo "=== VERIFICANDO NODE_MODULES ===" && \
  ls -la node_modules/@prisma/ 2>/dev/null || echo "@prisma no encontrado" && \
  exit 1)
```

Cómo Funciona

1. `set -o pipefail` → Si cualquier comando en un pipe falla, el pipe completo falla
2. `(npm run build | tee log)` → Ejecuta build y guarda output
3. `|| (error block)` → Si falla, ejecuta bloque de error
4. `tail -100` → Muestra últimas 100 líneas (más contexto que antes)
5. `ls -la node_modules/@prisma/` → Verifica que Prisma Client existe

Comparación

Aspecto	v8.8 (Bash)	v8.9 (sh)
Shell	Requiere Bash	Compatible con sh
PIPESTATUS	✅ Usa	❌ No usa (no existe)
pipefail	❌ No usa	✅ Usa
Compatibilidad	Solo Bash	sh, ash, dash, bash
Líneas de log	50	100
Verifica @prisma	❌ No	✅ Sí
Exit code	2 (error)	✅ Debería funcionar

Detalles Técnicos

Por Qué set -o pipefail

Sin pipefail:

```
false | true    # Exit code: 0 (éxito)
```

Con pipefail:

```
set -o pipefail
false | true    # Exit code: 1 (fallo)
```

En nuestro caso:

```
set -o pipefail
npm run build | tee log
# Si npm run build falla, el pipe completo falla
```

Estructura de Subshells

```
(comando1 | comando2) || (error_handler)
```

Si el subshell izquierdo falla, ejecuta el derecho.

Equivalente a:




```
try {
  npm run build | tee log
} catch {
  echo "ERROR"
  tail -100 log
  exit 1
}
```

Información de Debug Adicional

v8.9 agregaregistros:

```
=== VERIFICANDO NODE_MODULES ===
ls -la node_modules/@prisma/
```


Esto confirma si:

-  @prisma/client existe
-  Prisma generate se ejecutó
-  O si falta el módulo

Qué Veremos con v8.9

Si Falla el Build

```
=== BUILD NEXT.JS ===
[... npm run build output ...]

 BUILD FALLÓ
=== ÚLTIMAS 100 LÍNEAS DEL LOG ===

Creating an optimized production build
Failed to compile.

./app/lib/db.ts:1:0
Module not found: Can't resolve '@prisma/client'

=== ARCHIVOS EN /app ===
total 248
drwxr-xr-x  app/
drwxr-xr-x  node_modules/
-rw-r--r--  package.json
drwxr-xr-x  prisma/

=== VERIFICANDO NODE_MODULES ===
drwxr-xr-x  @prisma/client/
-rw-r--r--  package.json
[... o ...]
@prisma no encontrado
```

Si BUILD_ID No Se Crea

```

✓ Build completado exitosamente
=== VERIFICANDO BUILD ===
✗ ERROR: BUILD_ID no existe
=== CONTENIDO .next ===
[... contenido de .next ...]

```



Ventajas de v8.9

1. Compatible con Cualquier Shell

```

sh      ✓ (Alpine default)
ash     ✓ (BusyBox)
dash   ✓ (Debian/Ubuntu default)
bash   ✓ (retrocompatible)

```

2. Más Información de Debug

- 100 líneas en lugar de 50
- Verifica @prisma
- Muestra estructura de archivos
- Lista node_modules/@prisma/

3. Más Robusto

- `set -o pipefail` asegura captura de errores
- Subshells bien definidos
- No depende de características específicas de bash

4. Error Handling Claro

```

Build succeed → Verifica BUILD_ID → Success
                ↓ fail
                ERROR

Build fail → Show logs → Exit 1

```



Lo Que Necesito Ver

Cuando hagas rebuild con v8.9, debería:

1. ✓ **NO fallar con exit code 2** (error de sintaxis)
2. ✓ **Ejecutar npm run build**
3. ✓ **Capturar el output completo**
4. ✓ **Mostrar error detallado si falla**

Si falla, verás:

```

❌ BUILD FALLÓ
=== ÚLTIMAS 100 LÍNEAS DEL LOG ===
[... el error real del build ...]
=== ARCHIVOS EN /app ===
[... estructura de archivos ...]
=== VERIFICANDO NODE_MODULES ===
[... estado de @prisma ...]

```

Cópiame TODO ese output.

Posibles Escenarios

Escenario 1: Prisma Client No Encontrado

```

❌ BUILD FALLÓ
=== ÚLTIMAS 100 LÍNEAS ===
Module not found: Can't resolve '@prisma/client'

=== VERIFICANDO NODE_MODULES ===
@prisma no encontrado

```

Fix: Revisar stage de Prisma generate

Escenario 2: Error de TypeScript

```

❌ BUILD FALLÓ
=== ÚLTIMAS 100 LÍNEAS ===
Type error: Cannot find name 'User'
  at app/lib/types.ts:15:5

```

Fix: Corregir tipos o agregar ignoreBuildErrors

Escenario 3: Configuración Incorrecta

```

❌ BUILD FALLÓ
=== ÚLTIMAS 100 LÍNEAS ===
Error: Invalid next.config.js

```

Fix: Revisar next.config.js

Escenario 4: Variables de Entorno

```

❌ BUILD FALLÓ
=== ÚLTIMAS 100 LÍNEAS ===
Error: DATABASE_URL is required

```

Fix: Verificar ENV en Dockerfile

✓ Resumen

v8.8 → v8.9

```
- BUILD_EXIT_CODE=${PIPESTATUS[0]} # Solo bash
+ set -o pipefail                  # Compatible sh

- tail -50 /tmp/build-output.log   # Menos contexto
+ tail -100 /tmp/build-output.log  # Más contexto

+ ls -la node_modules/@prisma/    # Verifica Prisma
```

Estado Actual

```
v8.0-8.5 → Correcciones Prisma ✓
v8.6-8.7 → Standalone configurado ✓
v8.8      → Captura de errores (sintaxis bash)
v8.9      → ✓ CAPTURA COMPATIBLE CON SH
```

Progreso:

- ✓ Sintaxis shell corregida
- ✓ Compatible con Alpine sh
- ✓ Más información de debug
- 🔍 Listo para capturar error real

🎉 Próximo Paso

Rebuild con v8.9:

1. ✓ Script de shell funcionará (no más exit code 2)
2. ✓ Build de Next.js se ejecutará
3. ✓ Si falla, veremos el error completo
4. ✓ Con ese error, aplicaremos fix específico

Entonces:

- Si es error de Prisma → v8.10 fix Prisma
- Si es error de TypeScript → v8.10 fix TS
- Si es error de config → v8.10 fix config
- Si es otro → v8.10 fix específico

Versión: 8.9

Fecha: 2025-10-08 02:40 GMT

Estado: ✓ SH COMPATIBLE

Cambio principal:

- Script compatible con /bin/sh
- No requiere bash
- Captura 100 líneas de log
- Verifica node_modules/@prisma/

Objetivo: Eliminar exit code 2 y capturar error real de Next.js build.

¡Trigger rebuild y copia el output completo si falla! 🚀