## **Dockerfile v8.4 - Prisma Debug Avanzado**



### Problema Persistente

A pesar de las correcciones previas, el error continúa:

```
ERROR: process "npx prisma generate" exit code: 1
```



## X Debug Agregado en v8.4

### 1. Logs Completos del Schema

```
RUN echo "=== CONTENIDO SCHEMA ===" && \
    cat prisma/schema.prisma
```

Propósito: Ver el contenido exacto del schema que se está usando en el build.

#### 2. Verificación de Prisma CLI

```
RUN echo "=== VERIFICANDO PRISMA CLI ===" && \
    npx prisma --version
```

Propósito: Confirmar que Prisma CLI está instalado y funcional.

#### 3. DATABASE\_URL Temporal

```
RUN DATABASE URL="postgresql://user:pass@localhost:5432/db" npx prisma generate
```

Propósito: Prisma necesita una DATABASE URL válida durante generate, aunque no se conecte a la DB.



## 🔬 Qué Revelarán los Logs

### Escenario 1: Schema Corrupto

```
=== CONTENIDO SCHEMA ===
generator client {
  ... (contenido del schema)
```

**Acción:** Verificar sintaxis y configuración

### **Escenario 2: Prisma CLI No Disponible**

```
=== VERIFICANDO PRISMA CLI ===
command not found: prisma
```

Acción: Verificar instalación de dependencias

### **Escenario 3: Error de Binary Target**

```
Error: Generator failed:
Error: Unable to require(`<binary-path>`)
```

Acción: Ajustar binary Targets en schema

#### Escenario 4: Error de Sintaxis

```
Error: Prisma schema validation failed.
   --> schema.prisma:X
```

Acción: Corregir sintaxis en schema.prisma

## **III** Posibles Causas Restantes

### 1. Binary Target Incorrecto

Actual:

```
binaryTargets = ["native", "linux-musl-openssl-3.0.x"]
```

#### Alternativas a probar:

```
# Para Alpine x86_64
binaryTargets = ["native", "linux-musl-openssl-3.0.x"]

# Para Alpine ARM64
binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]

# Compatibilidad máxima
binaryTargets = [
    "native",
    "linux-musl-openssl-3.0.x",
    "linux-musl-arm64-openssl-3.0.x",
    "debian-openssl-3.0.x"
]
```

### 2. Dependencias Faltantes

Verificar en package.json:

```
"dependencies": {
    "@prisma/client": "6.7.0"
  "devDependencies": {
    "prisma": "6.7.0"
}
```

Ambos deben estar presentes y en la misma versión.

### 3. OpenSSL Version Mismatch

Alpine 3.18+ usa OpenSSL 3.0:

```
FROM node:18-alpine # Alpine 3.18
```

### Si usa Alpine anterior:

```
binaryTargets = ["native", "linux-musl-openssl-1.1.x"]
```

#### 4. Sintaxis del Schema

#### **Verificar:**

- Todos los modelos tienen sintaxis correcta
- Relaciones están bien definidas
- No hay campos duplicados
- Tipos de datos son válidos



## 🔧 Próximos Pasos Según los Logs

## Si los logs muestran el schema completo:

- 1. Copiar el schema del log
- 2. Verificar sintaxis en https://www.prisma.io/docs
- 3. Corregir cualquier error

### Si Prisma CLI no está disponible:

- 1. Verificar package.json incluye prisma en devDependencies
- 2. Confirmar que npm install completó exitosamente
- 3. Revisar stage de dependencies

### Si hay error de binary target:

- 1. Identificar la arquitectura del servidor
- 2. Ajustar binaryTargets en schema.prisma
- 3. Rebuild

## Si hay error de sintaxis:

- 1. Corregir línea específica indicada
- 2. Validar schema localmente

3. Commit y push



## 📝 Logs Esperados (Éxito)

```
=== CONTENIDO DEL DIRECTORIO ===
total 48
                       root
                                     4096 Oct 6 18:35 .
drwxr-xr-x 1 root
                                     4096 Oct 6 18:35 ..
2345 Oct 6 18:35 package.json
4096 Oct 6 18:35 prisma
drwxr-xr-x 1 root
                        root
-rw-r--r-- 1 root root
drwxr-xr-x 2 root root
=== PRISMA SCHEMA ===
total 12
-rw-r--r-- 1 root
                       root
                                     5678 Oct 6 18:35 schema.prisma
=== CONTENIDO SCHEMA ===
generator client {
    provider = "prisma-client-js"
    binaryTargets = ["native", "linux-musl-openssl-3.0.x"]
}
datasource db {
    provider = "postgresql"
    url = env("DATABASE URL")
}
model User {
=== VERIFICANDO PRISMA CLI ===
Query Engine (Node-API) : libquery-engine ... (at node_modules/...)
Migration Engine : migration-engine-cli ... (at node_modules/...)
Introspection Engine : introspection-core ... (at node_modules/...)
Format Binary : prisma-fmt ... (at node_modules/...)
Default Engines Hash : ...
Studio
                       : 0.xyz.0
=== GENERANDO CLIENTE PRISMA ===
Environment variables loaded from .env
Prisma schema loaded from prisma/schema.prisma
Generated Prisma Client (v6.7.0) to ./node_modules/@prisma/client

✓ Cliente Prisma generado
```

# **6** Objetivo de v8.4

Obtener información detallada sobre qué está fallando exactamente en npx prisma generate.

#### Con estos logs, podremos:

- 1. Ver el schema completo usado en el build
- 2. Confirmar disponibilidad de Prisma CLI
- 3. V Identificar errores específicos
- 4. <a> Hacer correcciones precisas</a>

Versión: 8.4

Fecha: 2025-10-06 18:35 GMT

Estado: Q MODO DEBUG ACTIVADO

#### **Cambios:**

Add: Logs detallados del schemaAdd: Verificación de Prisma CLI

Add: DATABASE\_URL temporal para generateMaintain: Todas las optimizaciones previas

**Próximo paso:** Trigger rebuild y revisar logs completos de EasyPanel para identificar la causa exacta del error.