

FIX CRÍTICO - Prisma Output Path Corregido

PROBLEMA IDENTIFICADO

Error en build:

```
Module not found: Can't resolve '.prisma/client/index-browser'

Import trace for requested module:
./components/clients/personal-references-form.tsx
./app/admin/clients/[id]/page.tsx
```

Causa raíz:

El `generator client` en `prisma/schema.prisma` tenía un `output` hardcodedo con la ruta local:

```
generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]
  output = "/home/ubuntu/escalafin_mvp/app/node_modules/.prisma/client" # 
PROBLEMA
}
```

Esta ruta funciona en **local** pero **falla en Docker** porque:

- En local: `/home/ubuntu/escalafin_mvp/app/node_modules/.prisma/client`
- En Docker: `/app/node_modules/.prisma/client`

SOLUCIÓN APLICADA

Commit: 38f9ed6

Fecha: 27 de octubre de 2025

1. Corregir Prisma Schema

Archivo: `app/prisma/schema.prisma`

Antes:

```
generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]
  output = "/home/ubuntu/escalafin_mvp/app/node_modules/.prisma/client"
}
```

Después:

```
generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]
  # output removido - usa ubicación por defecto
}
```

2. Mejorar Dockerfile

Cambios en el Dockerfile:

1. Usar `yarn prisma generate` en lugar de `npx prisma generate` :

- Mantiene consistencia con el lockfile
- Evita descargas innecesarias de paquetes

2. Agregar verificación estricta:

```
dockerfile
if [ -d "node_modules/.prisma/client" ]; then
  echo "✅ Directorio node_modules/.prisma/client encontrado"
else
  echo "❌ ERROR: Directorio node_modules/.prisma/client NO encontrado"
  exit 1
fi
```

POR QUÉ FUNCIONABA ANTES Y AHORA NO

En el commit anterior (c2804ba), corregimos el error de sintaxis en `start.sh` , pero ese cambio limpió el cache de build en EasyPanel. Al hacer rebuild con cache limpio, el problema del `output` hardcodeado en Prisma se manifestó.

Secuencia de eventos:

1. ✅ Local: funciona con `output` hardcodeado (ruta correcta)
2. 🟡 Docker (con cache): funcionaba porque el cliente ya estaba generado
3. ❌ Docker (sin cache): falla porque intenta generar en ruta incorrecta

INSTRUCCIONES PARA DESPLEGAR

Paso 1: Verificar Commit en GitHub

```
cd /home/ubuntu/escalafin_mvp
git log --oneline -n 1
```

Deberías ver:

```
38f9ed6 fix: Eliminar output hardcodeado en Prisma schema
```

Paso 2: Limpiar Cache en EasyPanel

1. Ve a tu servicio en EasyPanel

2. Settings → **Clear Build Cache**

3. Esto es necesario para que regenere el Prisma Client correctamente

Paso 3: Rebuild

1. Deployments → **Deploy / Rebuild**

2. Espera a que termine el build (3-5 minutos)

Paso 4: Verificar Logs de Build

Durante el build, deberías ver:

```

🔧 Limpiando y generando Prisma Client...
✅ Prisma Client generado
📋 Verificando tipos generados...
✅ Directorio node_modules/.prisma/client encontrado
total 1234
-rw-r--r-- 1 root root ... index.d.ts
-rw-r--r-- 1 root root ... index.js
...

🏗 Building Next.js...
✅ Build completado
```

NO deberías ver:

```

❌ Module not found: Can't resolve '.prisma/client/index-browser'
```

CHECKLIST DE VERIFICACIÓN

- [] Commit `38f9ed6` confirmado en GitHub
- [] Cache de build limpiado en EasyPanel
- [] Rebuild iniciado
- [] Build completado sin errores de Prisma
- [] Aplicación iniciada correctamente
- [] Health check verde

SI EL PROBLEMA PERSISTE

Verificar que el Schema Esté Correcto

```

cd /home/ubuntu/escalafin_mvp/app
grep -A 3 "generator client" prisma/schema.prisma
```

Debe mostrar (SIN línea `output`):

```
generator client {
  provider = "prisma-client-js"
  binaryTargets = ["native", "linux-musl-arm64-openssl-3.0.x"]
}
```

Regenerar Localmente (Opcional)

Si quieres verificar que funciona localmente:

```
cd /home/ubuntu/escalafin_mvp/app
rm -rf node_modules/.prisma node_modules/@prisma/client
yarn prisma generate
ls -la node_modules/.prisma/client/
```



ARCHIVOS MODIFICADOS

1. **app/prisma/schema.prisma**

- Removido: `output = "/home/ubuntu/escalafin_mvp/app/node_modules/.prisma/client"`
- Ahora usa ubicación por defecto

2. **Dockerfile**

- Cambiado: `npx prisma generate` → `yarn prisma generate`
- Agregado: verificación con `exit 1` si falla



RESULTADO ESPERADO

Con este fix, tu aplicación debería:

1. Generar Prisma Client correctamente en Docker
2. Compilar Next.js sin errores de módulos
3. Funcionar en local y en Docker con el mismo schema
4. No depender de rutas hardcodeadas



ORDEN DE FIXES APLICADOS

1. **Commit** `c2804ba` : Fix sintaxis en start.sh (fi duplicado)
2. **Commit** `38f9ed6` : Fix Prisma output path hardcodeado ← **ACTUAL**

Ambos fixes son necesarios para un deployment exitoso.

Fecha: 27 de octubre de 2025

Autor: DeepAgent

Commit: 38f9ed6

Estado: LISTO PARA DEPLOY