



Fix: Prisma WASM Files Missing



Problema Identificado

Error en Runtime:

```
Error: ENOENT: no such file or directory, open '/app/node_modules/.bin/prisma_schema_build_bg.wasm'
```

Análisis:

- Prisma CLI se estaba copiando correctamente
- Los archivos WASM asociados NO se copiaban
- Prisma necesita `prisma_schema_build_bg.wasm` para funcionar
- El Dockerfile solo copiaba el ejecutable, no todo el directorio `.bin`



Causa Raíz

El Dockerfile tenía:

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

Esto solo copiaba el archivo ejecutable `prisma`, pero **NO** los archivos WASM que están en el mismo directorio:

- `prisma_schema_build_bg.wasm`
- `prisma_fmt_build_bg.wasm`
- `query-engine.js`
- Otros archivos necesarios



Solución Aplicada

Cambio en Dockerfile

ANTES:

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

DESPUÉS:

```
# Copy ENTIRE .bin directory to include all Prisma WASM files
COPY --from=builder /app/node_modules/.bin ./node_modules/.bin
```

Razón:

- Copia TODO el directorio `.bin` (no solo el ejecutable)
- Incluye todos los archivos WASM necesarios
- Incluye cualquier otro binario que Prisma pueda necesitar
- Es la forma más robusta y a prueba de futuras actualizaciones






Archivos Incluidos Ahora

Con este cambio, se copian:

```
node_modules/.bin/
├── prisma                ← Ejecutable principal
├── prisma_schema_build_bg.wasm ← FALTABA (causa del error)
├── prisma_fmt_build_bg.wasm  ← Formatter WASM
└── [otros binarios si existen]
```





Copias de Prisma Completas

El Dockerfile ahora copia:





1.  `prisma/` - Schema files
2.  `node_modules/.prisma/` - Generated Prisma Client
3.  `node_modules/.bin/` - **CLI + WASM files (FIX APLICADO)**
4.  `node_modules/prisma/` - Prisma package
5.  `node_modules/@prisma/` - Prisma modules

Impacto

Antes del Fix:

-  Prisma CLI falla al intentar usar WASM
-  `prisma db push` no se ejecuta
-  Tablas no se crean en la base de datos
-  App falla con "relation does not exist"

Después del Fix:

-  Prisma CLI funcionará correctamente
-  `prisma db push` se ejecutará exitosamente
-  Tablas se crearán automáticamente
-  App iniciará sin errores de DB

Testing Plan

1. Verificar en Build

Durante el build, verificar que los archivos se copian:

```
# En el stage runner, agregar verificación:
RUN ls -la node_modules/.bin/ && \
    test -f node_modules/.bin/prisma_schema_build_bg.wasm || \
    (echo "❌ WASM file missing" && exit 1)
```

2. Verificar en Runtime

Los logs mostrarán:

```

🔄 Sincronizando esquema con base de datos...
📌 Usando comando: node_modules/.bin/prisma
🚀 Ejecutando: node_modules/.bin/prisma db push --accept-data-loss --skip-generate
✅ Esquema sincronizado exitosamente

```

3. Verificar Tablas Creadas

```

# Conectar a PostgreSQL y verificar:
\d
# Debe mostrar todas las tablas: users, clients, loans, etc.

```

Próximos Pasos

1. Commit y Push:

```

bash
git add Dockerfile FIX_PRISMA_WASM_FILES.md
git commit -m "fix(docker): copiar archivos WASM de Prisma"
git push origin main

```

2. Deploy en EasyPanel:

- Pull del nuevo commit
- Limpiar caché (obligatorio)
- Rebuild completo
- Verificar logs de sincronización de schema

3. Verificar Éxito:

- ✅ Logs muestran "Esquema sincronizado exitosamente"
- ✅ App inicia sin errores de "relation does not exist"
- ✅ Login funciona correctamente
- ✅ Base de datos tiene todas las tablas

Referencias

- **Prisma WASM Issue:** <https://github.com/prisma/prisma/issues/12649>
- **Docker Best Practices:** Copiar directorios completos en lugar de archivos individuales
- **CitaPlanner Reference:** Este patrón está validado en producción


✅ Resultado Esperado


Este fix debería resolver **definitivamente** el problema de las tablas faltantes porque:

1. ✅ Prisma CLI tendrá todos los archivos necesarios
2. ✅ `prisma db push` podrá ejecutarse correctamente
3. ✅ Las tablas se crearán en la primera ejecución
4. ✅ La aplicación funcionará end-to-end

Fecha: 28 Octubre 2025

Commit: Pendiente

Status:  Fix aplicado, listo para deploy

Prioridad:  CRÍTICO - Resuelve el problema raíz