



Fix: Debug Mejorado para Prisma DB Push



Problema Actual

Los logs de EasyPanel muestran:

```
ERROR: relation "public.users" does not exist at character 74
```

Esto indica que:

- ☒ La aplicación se inicia correctamente
- ☒ La conexión a la base de datos funciona
- ☒ Las tablas NO se están creando
- ☒ El comando `prisma db push` no se ejecuta o falla silenciosamente



Mejoras Aplicadas

1. Script de Inicio Mejorado (`start-improved.sh`)

Logging Detallado:

```
echo " 🔴 Usando comando: $PRISMA_CMD"
echo " 🔴 Working directory: $(pwd)"
echo " 🔴 Schema location: $(pwd)/prisma/schema.prisma"
```

Captura de Output Completo:

```
DB_PUSH_OUTPUT=$(($PRISMA_CMD db push --accept-data-loss --skip-generate 2>&1)
DB_PUSH_EXIT_CODE=$?

echo " 📋 Output completo del comando:"
echo "$DB_PUSH_OUTPUT"
```



Verificación del Schema:

```
if [ ! -f "prisma/schema.prisma" ]; then
  echo " 🔴 ERROR: prisma/schema.prisma no encontrado"
  exit 1
fi
```

2. Nuevo Script de Debug (`start-debug.sh`)

Script especializado para debugging que verifica:

- ☒ Directorio de trabajo actual
- ☒ Versiones de Node y NPM
- ☒ Disponibilidad de Prisma CLI
- ☒ Existencia de DATABASE_URL
- ☒ Existencia de schema.prisma

-  Conexión a la base de datos
-  Output completo de `db push`

Uso:

```
# En el Dockerfile, cambiar temporalmente:  
CMD ["dumb-init", "sh", "/app/start-debug.sh"]
```



Próximos Pasos para Diagnóstico

Opción 1: Deploy con Logging Mejorado

1. Pull del próximo commit (con start-improved.sh mejorado)
2. Rebuild en EasyPanel
3. **Revisar logs completos** que ahora mostrarán:
 - Comando de Prisma usado
 - Working directory
 - Output completo de db push
 - Código de salida


Opción 2: Deploy con Script de Debug

1. Cambiar el CMD en Dockerfile a `start-debug.sh`
2. Rebuild en EasyPanel
3. Logs mostrarán diagnóstico completo paso a paso




Qué Buscar en los Logs

Escenario 1: Schema No Encontrado

 ERROR: prisma/schema.prisma no encontrado

Solución: Verificar que el schema se copie correctamente en el Dockerfile

Escenario 2: Prisma CLI No Disponible

 Prisma CLI no encontrado

Solución: Verificar la copia de node_modules/.bin/prisma en Dockerfile

Escenario 3: Error de Conexión a DB

Error: Can't reach database server at...

Solución: Verificar DATABASE_URL en variables de entorno de EasyPanel

Escenario 4: Prisma db push Falla

✖ ERROR: db push falló con código: 1
📋 Output completo del comando:
[Aquí veremos el error real]

Solución: Depende del error específico mostrado



Archivos Modificados

1. **start-improved.sh**
 - Agregado: Logging detallado de todas las operaciones
 - Agregado: Captura completa de output de db push
 - Agregado: Verificación de existencia de schema.prisma
 - Mejorado: Manejo de errores con mensajes claros
2. **start-debug.sh** (NUEVO)
 - Script especializado para debugging
 - Verifica cada prerequisite paso a paso
 - Output muy verbose para diagnóstico completo



Commit y Deploy

```
cd /home/ubuntu/escalafin_mvp
git add start-improved.sh start-debug.sh FIX_DB_PUSH_DEBUG.md
git commit -m "fix(prisma): mejorar logging y debugging de db push"
git push origin main
```

Luego en EasyPanel:

1. Pull del último commit
2. Limpiar caché
3. Rebuild
4. **Revisar logs cuidadosamente** para ver output detallado

Fecha: 28 Oct 2025

Status: ⌚ Pendiente deploy con logging mejorado

Objetivo: Identificar por qué db push no crea las tablas