

Guía de Troubleshooting - Problemas de Conexión a PostgreSQL

Tabla de Contenidos

1. [Diagnóstico del Problema](#)
 2. [Causas Posibles](#)
 3. [Verificación en Easypanel](#)
 4. [Soluciones Paso a Paso](#)
 5. [Comandos Útiles para Debugging](#)
 6. [Ejemplos de Configuración Correcta](#)
 7. [Prevención de Problemas Futuros](#)
-

Diagnóstico del Problema

Síntomas Observados

Basado en los logs de despliegue, se identificaron los siguientes problemas:

```
[2025-10-06 01:08:00] ⚠ Intento 1/30 - Esperando conexión a la base de datos...  
...  
[2025-10-06 01:09:00] ❌ No se pudo conectar a la base de datos después de 30 intentos  
[2025-10-06 01:09:00] ❌ No se pudo establecer conexión con la base de datos
```

Error Principal:

```
PrismaClientKnownRequestError: The table public.master_admin_config does not exist  
in the current database.
```

Análisis del Problema

Componente	Estado	Descripción
Servicio PostgreSQL	✗ Inaccesible	El contenedor no puede resolver el hostname <code>cloud-mx_citaplanner-db</code>
Conexión de Red	✗ Fallida	30 intentos de conexión fallidos en 60 segundos
Migraciones Prisma	✗ No Ejecutadas	Sin conexión a BD, las migraciones no se aplicaron
Tabla <code>master_admin_config</code>	✗ No Existe	Depende de las migraciones
Aplicación Next.js	⚠ Iniciada	Se inició pero sin funcionalidad de BD

Problema Raíz

La aplicación no puede conectarse al servicio PostgreSQL, lo que impide:

1. Ejecutar las migraciones de Prisma
2. Crear las tablas necesarias
3. Acceder a la funcionalidad de base de datos

Causas Posibles

1. Servicio PostgreSQL No Creado o No Iniciado

Probabilidad: ALTA ●

- El servicio de base de datos no existe en Easypanel
- El contenedor PostgreSQL no está corriendo
- El servicio se detuvo inesperadamente

Cómo Verificar:

- Ir a Easypanel → Proyecto → Verificar si existe el servicio de base de datos
- Revisar el estado del servicio (running/stopped/error)

2. Nombre de Red o Hostname Incorrecto

Probabilidad: ALTA ●

- El hostname `cloudmx_citaplanner-db` no está correctamente configurado
- Los servicios no están en la misma red de Docker
- Error tipográfico en el nombre del servicio

Configuración Actual:

```
DATABASE_URL=postgres://postgres:674a351a07db86883d92@cloudmx_citaplanner-db:5432/citaplanner-db?sslmode=disable
```

Cómo Verificar:

- El nombre del servicio debe coincidir EXACTAMENTE con el nombre en Easypanel
- Los servicios deben estar en el mismo proyecto/red

3. Orden de Inicio Incorrecto

Probabilidad: MEDIA 🟡

- La aplicación inicia antes que PostgreSQL esté listo
- PostgreSQL está iniciando pero aún no acepta conexiones

Nota: El script `docker-entrypoint.sh` ya implementa 30 reintentos con 2 segundos de espera, lo que debería ser suficiente.

4. Credenciales Incorrectas

Probabilidad: BAJA 🟢

- Usuario o contraseña incorrectos
- Base de datos no existe

Nota: Este error aparecería DESPUÉS de establecer la conexión, no antes.

5. Puerto Incorrecto o Bloqueado

Probabilidad: BAJA 🟢

- Puerto 5432 no está disponible
- Firewall bloqueando el puerto

Nota: En redes internas de Docker, esto es poco probable.

✅ Verificación en Easypanel

Paso 1: Verificar que el Servicio PostgreSQL Existe

1. Acceder a Easypanel:

- Ir a `https://easypanel.io` (o tu instalación)
- Iniciar sesión


2. Navegar al Proyecto:

- Seleccionar el proyecto `cloudmx` (o el nombre de tu proyecto)

3. Buscar el Servicio de Base de Datos:

- Buscar un servicio llamado `citaplanner-db` o similar
- Verificar que existe y está en estado **Running** ✅

4. Si NO existe el servicio:

 ACCIÓN REQUERIDA: Crear el servicio PostgreSQL

Paso 2: Verificar la Configuración del Servicio PostgreSQL

1. Hacer clic en el servicio PostgreSQL


2. Verificar Configuración:

yaml

Service Name: citaplanner-db ← Debe coincidir con el hostname en DATABASE_URL

Image: postgres:15 (o similar)

Port: 5432

Status: Running 

3. Verificar Variables de Entorno:

env

POSTGRES_USER=postgres

POSTGRES_PASSWORD=674a351a07db86883d92 ← Debe coincidir con DATABASE_URL

POSTGRES_DB=citaplanner-db ← Debe coincidir con DATABASE_URL

4. Verificar Volúmenes:

- Debe tener un volumen persistente para /var/lib/postgresql/data
- Esto asegura que los datos no se pierdan al reiniciar

Paso 3: Verificar la Red del Proyecto

1. En Easypanel, ir a Project Settings o Network:

2. Verificar que ambos servicios están en la misma red:

...

Red del Proyecto: cloudmx_network (o similar)


Servicios en la Red:

└─ citaplanner (aplicación Next.js)

└─ citaplanner-db (PostgreSQL)

...

1. Si no están en la misma red:

 ACCIÓN REQUERIDA: Configurar ambos servicios en la misma red

Paso 4: Verificar la Variable DATABASE_URL en la Aplicación

1. Ir al servicio CitaPlanner (aplicación Next.js)

2. Hacer clic en “Environment Variables” o “Settings”

3. Verificar DATABASE_URL:

env

DATABASE_URL=postgres://postgres:674a351a07db86883d92@cloudmx_citaplanner-db:5432/

citaplanner-db?sslmode=disable

4. Componentes de la URL:

...

postgres:// ← Protocolo

postgres ← Usuario (debe coincidir con POSTGRES_USER)

← Separador

674a351a07db86883d92 ← Contraseña (debe coincidir con POSTGRES_PASSWORD)

@ ← Separador

cloudmx_citaplanner-db ← HOSTNAME (debe ser EXACTAMENTE el nombre del servicio)

:5432 ← Puerto

/citaplanner-db ← Nombre de la base de datos (debe coincidir con POSTGRES_DB)

?sslmode=disable ← Parámetros adicionales

...

Soluciones Paso a Paso

Solución 1: Crear el Servicio PostgreSQL (Si No Existe)

Si el servicio PostgreSQL no existe en Easypanel:

1. En Easypanel, ir al proyecto `cloudmx`
2. Hacer clic en “Add Service” o “Create Service”
3. Seleccionar “Database” → “PostgreSQL”
4. Configurar el servicio:

yaml

Name: citaplanner-db

Version: 15 (o la versión que prefieras)

5. Configurar Variables de Entorno:

env

POSTGRES_USER=postgres

POSTGRES_PASSWORD=674a351a07db86883d92

POSTGRES_DB=citaplanner-db

6. Configurar Volumen:

Mount Path: /var/lib/postgresql/data

Volume Name: citaplanner-db-data

7. Hacer clic en “Create” o “Deploy”

8. Esperar a que el servicio esté en estado “Running” (puede tomar 1-2 minutos)

Solución 2: Corregir el Hostname en DATABASE_URL

Si el nombre del servicio no coincide:

1. Ir al servicio PostgreSQL y copiar su nombre exacto
 - Ejemplo: `citaplanner-db`
2. Ir al servicio CitaPlanner (Next.js)
3. Editar la variable `DATABASE_URL` :

Si el servicio se llama `citaplanner-db` :

```
env
```

```
DATABASE_URL=postgres://postgres:674a351a07db86883d92@citaplanner-db:5432/citaplanner-db?sslmode=disable
```

Si el servicio incluye el prefijo del proyecto:

```
env
```

```
DATABASE_URL=postgres://postgres:674a351a07db86883d92@cloudmx_citaplanner-db:5432/citaplanner-db?sslmode=disable
```

1. Guardar cambios
2. Re-desplegar la aplicación:
 - Hacer clic en "Redeploy" o "Restart"

Solución 3: Verificar y Corregir las Credenciales

Si las credenciales no coinciden:

1. Obtener las credenciales del servicio PostgreSQL:
 - Ir al servicio PostgreSQL
 - Copiar `POSTGRES_USER` , `POSTGRES_PASSWORD` , `POSTGRES_DB`
2. Actualizar `DATABASE_URL` en la aplicación:

```
env
```

```
DATABASE_URL=postgres://[USER]:[PASSWORD]@[HOSTNAME]:5432/[DATABASE]?sslmode=disable
```

3. Ejemplo:

```
env
```

```
DATABASE_URL=postgres://postgres:nueva_contraseña@citaplanner-db:5432/citaplanner-db?sslmode=disable
```

4. Guardar y Re-desplegar

Solución 4: Asegurar que Ambos Servicios Están en la Misma Red

Para servicios en Easypanel:

1. Los servicios en el mismo proyecto suelen estar automáticamente en la misma red
2. Verificar en Project Settings:
 - Ambos servicios deben aparecer en la misma red Docker

3. Si están en proyectos diferentes:

- Mover ambos servicios al mismo proyecto, O
- Configurar una red compartida manualmente (avanzado)

Solución 5: Ejecutar Migraciones Manualmente

Una vez que la conexión funcione, ejecutar las migraciones:

Opción A: Usando el Script de Diagnóstico

```
# Desde el contenedor de la aplicación
./scripts/diagnose-db.sh
```

Opción B: Usando Easypanel Console

1. En Easypanel, ir al servicio CitaPlanner

2. Hacer clic en “Console” o “Shell”

3. Ejecutar:

```
bash
cd /app
npx prisma migrate deploy
```

4. Verificar que las migraciones se aplicaron:

```
bash
npx prisma db seed
```

Opción C: Desde Tu Máquina Local

1. Configurar DATABASE_URL temporalmente:

```
bash
export DATABASE_URL="postgres://postgres:674a351a07db86883d92@[IP_PUBLICA_0_TUNNEL]:5432/citaplanner-db?sslmode=disable"
```

2. Ejecutar migraciones:

```
bash
npx prisma migrate deploy
```

Comandos Útiles para Debugging

Desde el Contenedor de la Aplicación (Easypanel Console)

```
# 1. Verificar conectividad básica
ping -c 4 citaplanner-db

# 2. Verificar si el puerto 5432 está abierto
nc -zv citaplanner-db 5432
# o
telnet citaplanner-db 5432

# 3. Verificar resolución DNS
nslookup citaplanner-db
# o
host citaplanner-db

# 4. Probar conexión con psql (si está instalado)
psql "$DATABASE_URL" -c "SELECT version();"

# 5. Ver variables de entorno
echo $DATABASE_URL

# 6. Probar conexión con Prisma
npx prisma db push --skip-generate

# 7. Ver estado de las migraciones
npx prisma migrate status

# 8. Listar tablas (si la conexión funciona)
npx prisma db execute --stdin <<< "SELECT tablename FROM pg_tables WHERE schemaname='public';"
```

Desde el Contenedor PostgreSQL (Easypanel Console)

```
# 1. Verificar que PostgreSQL está corriendo
ps aux | grep postgres

# 2. Ver logs de PostgreSQL
tail -f /var/lib/postgresql/data/log/postgresql-*.log

# 3. Conectarse a la base de datos
psql -U postgres -d citaplanner-db

# 4. Dentro de psql:
\l                # Listar bases de datos
\c citaplanner-db # Conectar a la base de datos
\dt              # Listar tablas
\dt public.*     # Listar tablas en schema public
\d master_admin_config # Describir tabla específica
SELECT version(); # Ver versión de PostgreSQL
\q              # Salir
```


Usando el Script de Diagnóstico

```
# Ejecutar el script
cd /app
./scripts/diagnose-db.sh

# El script verificará:
# - Conectividad de red
# - Conexión a PostgreSQL
# - Estado de las migraciones
# - Tablas existentes
# - Configuración de la base de datos
```



Ejemplos de Configuración Correcta

Configuración en Easypanel

Servicio PostgreSQL

```
name: citaplanner-db
image: postgres:15
environment:
  POSTGRES_USER: postgres
  POSTGRES_PASSWORD: 674a351a07db86883d92
  POSTGRES_DB: citaplanner-db
volumes:
  - name: citaplanner-db-data
    mountPath: /var/lib/postgresql/data
ports:
  - containerPort: 5432
network: cloudmx_network
```

Servicio CitaPlanner (Next.js)

```
name: citaplanner
image: ghcr.io/tu-usuario/citaplanner:latest
environment:
  DATABASE_URL: postgres://postgres:674a351a07db86883d92@citaplanner-db:5432/
citaplanner-db?sslmode=disable
  MASTER_ADMIN_HASH: $2b$10$P/AV363LeWhZGK0kkr0N3eGmAlkmiTHKuzZzDKCAppFV.0Gzf0Za0
  NODE_ENV: production
ports:
  - containerPort: 3000
  publicPort: 80
network: cloudmx_network
dependsOn:
  - citaplanner-db
```

DATABASE_URL - Ejemplos

Formato Correcto

```
# Formato general
postgres://[usuario]:[contraseña]@[hostname]:[puerto]/[database]?[parametros]

# Ejemplo 1: Servicio simple
DATABASE_URL=postgres://postgres:mypassword@citaplanner-db:5432/citaplanner-db?
sslmode=disable

# Ejemplo 2: Con prefijo de proyecto
DATABASE_URL=postgres://postgres:mypassword@cloudmx_citaplanner-db:5432/citaplanner-
db?sslmode=disable

# Ejemplo 3: Con SSL
DATABASE_URL=postgres://postgres:mypassword@citaplanner-db:5432/citaplanner-db?
sslmode=require

# Ejemplo 4: Con pool de conexiones
DATABASE_URL=postgres://postgres:mypassword@citaplanner-db:5432/citaplanner-db?
sslmode=disable&connection_limit=10
```

Errores Comunes

```
# ✗ INCORRECTO: localhost (no funciona en Docker)
DATABASE_URL=postgres://postgres:password@localhost:5432/db

# ✗ INCORRECTO: 127.0.0.1 (no funciona en Docker)
DATABASE_URL=postgres://postgres:password@127.0.0.1:5432/db

# ✗ INCORRECTO: Hostname equivocado
DATABASE_URL=postgres://postgres:password@wrong-hostname:5432/db

# ✗ INCORRECTO: Caracteres especiales sin escapar
DATABASE_URL=postgres://postgres:p@ssw0rd!@citaplanner-db:5432/db

# ✓ CORRECTO: Caracteres especiales codificados
DATABASE_URL=postgres://postgres:p%40ssw0rd%21@citaplanner-db:5432/db
```



Prevención de Problemas Futuros

1. Health Checks

Agregar health checks en la configuración de Easypanel:

```
healthCheck:
  test: ["CMD", "pg_isready", "-U", "postgres"]
  interval: 10s
  timeout: 5s
  retries: 5
```

2. Dependencias de Servicios

Asegurar que la aplicación espere a PostgreSQL:

dependsOn:
- citaplanner-db

3. Logs y Monitoreo

- Revisar logs regularmente en Easypanel
- Configurar alertas para errores de conexión
- Monitorear uso de recursos de PostgreSQL

4. Backups Automáticos

- Configurar backups automáticos en Easypanel
- Probar la restauración periódicamente
- Mantener múltiples versiones de backup

5. Testing de Conectividad

Ejecutar el script de diagnóstico después de cada despliegue:

```
./scripts/diagnose-db.sh
```

Soporte Adicional

Si después de seguir esta guía el problema persiste:

1. Recopilar Información:

```
```bash
Ejecutar el script de diagnóstico
./scripts/diagnose-db.sh > diagnostico.txt 2>&1

Capturar logs
En Easypanel: Service → Logs → Download
```
```

1. Verificar Configuración:

- Captura de pantalla de la configuración del servicio PostgreSQL
- Captura de pantalla de las variables de entorno de CitaPlanner
- Captura de pantalla de la configuración de red

2. Contactar Soporte:

- Incluir el archivo `diagnostico.txt`
- Incluir capturas de pantalla
- Describir los pasos ya realizados

Checklist Final

Antes de considerar el problema resuelto, verificar:

- [] Servicio PostgreSQL existe y está en estado “Running”
- [] DATABASE_URL tiene el hostname correcto

- [] Credenciales coinciden entre servicios
 - [] Ambos servicios están en la misma red
 - [] Script de diagnóstico ejecuta sin errores
 - [] Migraciones de Prisma aplicadas exitosamente
 - [] Tabla `master_admin_config` existe
 - [] Endpoint `/api/admin/master/test-hash` funciona
 - [] Login en `/admin/master` funciona correctamente
-

Última actualización: 2025-10-06

Versión: 1.0.0