





Solución: Error de Conexión a Base de Datos en Vercel

Fecha: 2 de octubre de 2025

Error: P1001: Can't reach database server

Progreso Hasta Ahora

-  yarn.lock corregido (problema resuelto)
-  Dependencias instaladas correctamente
-  Prisma Client generado exitosamente
-  Error al conectar a la base de datos para migraciones

Error Actual

```
Error: P1001: Can't reach database server at
ep-cool-sea-a5jt7r2q-pooler.us-east-2.aws.neon.tech:5432

Please make sure your database server is running at
ep-cool-sea-a5jt7r2q-pooler.us-east-2.aws.neon.tech:5432.
```

Causa del Problema

Neon Database con Pooling:

- Neon usa `pgbouncer` para pooling de conexiones
- El endpoint con `-pooler` funciona para queries de la aplicación
- **PERO** no funciona bien para migraciones de Prisma
- Las migraciones necesitan conexión **directa** sin pooler

Solución:

Usar **DOS connection strings**:

1. **Con pooler** → Para la aplicación (queries normales)
2. **Sin pooler** → Para migraciones (DDL operations)

Cambios Realizados

1. Actualizado `schema.prisma`

He actualizado el archivo de Prisma para soportar ambas URLs:

```
datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")           // Con pooler (app)
  directUrl = env("DIRECT_DATABASE_URL")   // Sin pooler (migraciones)
}
```

Commit: 04c42fe - Ya está en GitHub 

Configurar Variables de Entorno en Vercel

Paso 1: Obtener las Connection Strings de Neon

1. Ve a: <https://console.neon.tech>
2. Inicia sesión y selecciona tu proyecto: **hotel_pms_production**
3. Ve a la sección **“Connection Details”** o **“Dashboard”**
4. Busca el menú desplegable que dice **“Connection string”**

Paso 2: Copiar Ambas URLs




Necesitas copiar **DOS connection strings** diferentes:

A. Connection String con Pooler (para DATABASE_URL)

Selecciona: **“Pooled connection”** o busca la URL que incluye `-pooler`:

```
postgresql://neondb_owner:TU_PASSWORD@ep-cool-sea-a5jt7r2q-pooler.us-east-2.aws.neon.tech/hotel_pms_production?sslmode=require
```

Características:




-  Incluye `-pooler` en el hostname
-  Usa el puerto 5432
-  Óptima para queries de la aplicación

B. Connection String Directa (para DIRECT_DATABASE_URL)

Selecciona: **“Direct connection”** o busca la URL que NO incluye `-pooler`:

```
postgresql://neondb_owner:TU_PASSWORD@ep-cool-sea-a5jt7r2q.us-east-2.aws.neon.tech/hotel_pms_production?sslmode=require
```

Características:

-  NO incluye `-pooler` en el hostname
-  Usa el puerto 5432
-  Necesaria para migraciones

Configurar en Vercel

Paso 3: Ir a Environment Variables

1. Ve a tu proyecto en Vercel Dashboard

2. Haz clic en **“Settings”**
3. En el menú lateral, haz clic en **“Environment Variables”**

Paso 4: Agregar/Actualizar Variables

Necesitas configurar **5 variables** en total:

Variable	Valor	Descripción
DATABASE_URL	postgresql:// neondb_owner:PASSWORD@ep- cool-sea-a5jt7r2q-pool- er.us-east-2.aws.neon.tech/ hotel_pms_production?sslmode=require	Con pooler (app)
DIRECT_DATABASE_URL	postgresql:// neondb_owner:PASSWORD@ep- cool-sea-a5jt7r2q.us- east-2.aws.neon.tech/ hotel_pms_production?sslmode=require	Sin pooler (migraciones)
NEXTAUTH_URL	https://hotel-pms-paseo- las-mercedes.vercel.app	URL de la app (actualizar después)
NEXTAUTH_SECRET	aG8t3lP@s30L@sM3rc3d3s! 2024#PMS\$V3rc3l%S3cr3t&K3y* S3cur3!	Secret para NextAuth
NODE_ENV	production	Ambiente de producción

⚠ IMPORTANTE:

- Reemplaza `PASSWORD` con tu contraseña real de Neon
- La única diferencia entre `DATABASE_URL` y `DIRECT_DATABASE_URL` es `-pooler` en el hostname

Paso 5: Configurar los Ambientes

Para cada variable, asegúrate de marcar:

- ☒ **Production**
- ☒ **Preview**
- ☒ **Development**



Re-deployar

Una vez configuradas las variables:

Opción A: Deployment Automático

- Vercel detectará el nuevo commit (04c42fe)

- Inicialará un nuevo deployment automáticamente

Opción B: Deployment Manual

1. Ve a **“Deployments”** en Vercel Dashboard
2. Haz clic en el botón **“Redeploy”**
3. Selecciona: **“Use existing build cache: No”**
4. Haz clic en **“Redeploy”**

✓ Lo que Debería Suceder Ahora

El nuevo build ejecutará:

- ✓ yarn install
- ✓ npx prisma generate
- ✓ npx prisma migrate deploy
 - ↳ Usa DIRECT_DATABASE_URL (sin pooler)
 - ↳ Crea todas las tablas en Neon
- ✓ yarn build
 - ↳ La app usa DATABASE_URL (con pooler)
- ✓ Deploy completado 🎉

🔍 Verificar en los Logs

Durante el build, busca estas líneas:

```
✓ Generated Prisma Client
Datasource "db": PostgreSQL database "hotel_pms_production"...
Applying migration `20250810115033_initial_with_housekeeping`
Applying migration `20250810141625_add_guest_communications_and_templates`
✓ Migrations applied successfully
```

📊 Verificar Tablas Creadas

Después del deployment exitoso:

1. Ve a: <https://console.neon.tech>
2. Selecciona: **hotel_pms_production**
3. Ve a: **“SQL Editor”**
4. Ejecuta:

```
SELECT table_name
FROM information_schema.tables
WHERE table_schema = 'public'
ORDER BY table_name;
```

Deberías ver todas estas tablas:

```

Account
CommunicationTemplate
Guest
GuestCommunication
HousekeepingAssignment
HousekeepingTask
MaintenanceRequest
Payment
Reservation
Room
RoomServiceOrder
ServiceItem
Session
Staff
User
VerificationToken
... (y más)

```

Troubleshooting

Si sigue sin conectar:

1. Verificar que la Base de Datos esté Activa

En Neon Console:

- Verifica que el proyecto **no esté suspendido**
- Verifica que el **Compute endpoint** esté en estado “Active”
- Si está suspendido, actívalo

2. Verificar las Contraseñas

- Asegúrate de usar la **misma contraseña** en ambas URLs
- Verifica que no haya espacios extras al copiar/pegar
- Si no estás seguro, **genera una nueva contraseña** en Neon:
 1. Ve a Neon Console → tu proyecto
 2. Ve a “Connection Details”
 3. Haz clic en “Reset password”
 4. Usa la nueva contraseña en ambas variables

3. Verificar los Hostnames

✓ CORRECTO (con pooler):
ep-cool-sea-a5jt7r2q-pooler.us-east-2.aws.neon.tech

✓ CORRECTO (sin pooler):
ep-cool-sea-a5jt7r2q.us-east-2.aws.neon.tech

✗ INCORRECTO:
Las URLs deben terminar en .aws.neon.tech

4. Verificar los Parámetros de Conexión

Ambas URLs deben incluir:

```
?sslmode=require
```

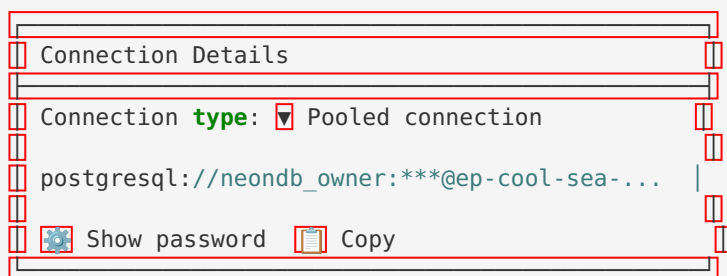
No deben incluir (Prisma lo agrega automáticamente si es necesario):

```
&pgbouncer=true&connect_timeout=10
```



Captura de Ejemplo - Neon Console

En Neon, verás algo como:



Cambia a: "Direct connection" para la segunda URL



Checklist de Configuración

Antes de re-deployar, verifica:

- [] DATABASE_URL configurada en Vercel (con `-pooler`)
- [] DIRECT_DATABASE_URL configurada en Vercel (sin `-pooler`)
- [] Ambas URLs tienen la **misma contraseña**
- [] Ambas URLs incluyen `?sslmode=require`
- [] NEXTAUTH_URL configurada
- [] NEXTAUTH_SECRET configurada
- [] NODE_ENV = `production`
- [] Todas las variables marcadas para Production/Preview/Development
- [] Commit `04c42fe` está en GitHub
- [] Base de datos Neon está activa



Resumen del Problema y Solución

Problema:

pgbouncer (pooler) no soporta DDL operations (CREATE TABLE, etc.)
→ Las migraciones fallan

Solución:

Usar dos URLs diferentes:

- ✓ DATABASE_URL (con pooler) → App queries
- ✓ DIRECT_DATABASE_URL (sin pooler) → Migraciones

Implementación:

1. Actualizar schema.prisma con directUrl ✓
2. Configurar ambas variables en Vercel 🔗
3. Re-deployar 🚀



Nota sobre Pooling

¿Por qué dos URLs?

- **Pooler (pgbouncer):** Rápido, eficiente para queries, pero no soporta DDL
- **Direct:** Más lento, pero necesario para CREATE TABLE, ALTER TABLE, etc.
- **Best practice:** Usar pooler para app, direct solo para migraciones



Próximos Pasos Inmediatos

1. 🔑 **Obtener ambas connection strings de Neon**
2. 🗝️ **Configurar DATABASE_URL en Vercel** (con `-pooler`)
3. 🗝️ **Configurar DIRECT_DATABASE_URL en Vercel** (sin `-pooler`)
4. 🚀 **Re-deployar**
5. 👁️ **Monitorear los logs del build**
6. ✓ **Verificar tablas creadas en Neon**
7. 🎉 **Probar la aplicación**

Última actualización: 2 de octubre de 2025 - 16:30

Commit de solución: 04c42fe

Estado: Listo para re-deploy con nueva configuración