

# Instrucciones para Desplegar el Sistema de Pie Diabético

## ✓ PROBLEMA RESUELTO

Este archivo ya contiene las correcciones necesarias para que tu aplicación funcione correctamente en Vercel:

### Modificaciones incluidas:

- **package.json** actualizado con scripts que automáticamente crean las tablas de la base de datos
- Scripts de build: `prisma db push --accept-data-loss && prisma generate && next build`
- Script de Vercel: `vercel-build` con los mismos comandos

## 📦 CONTENIDO DEL ARCHIVO

PIE_DIABETICO_CLEAN/	
├── app/	# Tu aplicación Next.js
│   ├── package.json	# ✓ YA CORREGIDO con scripts de DB
│   ├── prisma/	# Esquemas de <b>base</b> de datos
│   ├── app/	# Páginas y rutas
│   ├── components/	# Componentes React
│   └── ...	# Otros archivos del proyecto
├── .env.example	# Variables de entorno de ejemplo
├── README.md	# Documentación
└── DEPLOYMENT.md	# Guía de despliegue

## 🚀 PASOS PARA ACTUALIZAR TU PROYECTO EN VERCEL

### Opción A: Reemplazar carpeta completa (RECOMENDADO)

1. **Extrae el archivo** `PIE_DIABETICO_LIMPIO_FINAL.tar.gz`
2. **Ve a tu repositorio de GitHub** (donde está tu proyecto)
3. **Elimina la carpeta** `app` existente
4. **Arrastra y suelta** la nueva carpeta `app` desde `PIE_DIABETICO_CLEAN/app/`
5. **Commit y push** los cambios

### Opción B: Solo actualizar package.json (RÁPIDO)

1. **Ve a tu repositorio de GitHub**
2. **Navega a** `app/package.json`
3. **Edita el archivo** y reemplaza la sección `"scripts"` con:

```
"scripts": {
  "dev": "next dev",
  "build": "prisma db push --accept-data-loss && prisma generate && next build",
  "start": "next start",
  "lint": "next lint",
  "postinstall": "prisma generate",
  "vercel-build": "prisma db push --accept-data-loss && prisma generate && next build"
}
```

1. **Guarda y commit** los cambios

## ¿QUÉ PASARÁ DESPUÉS?





1. **Vercel detectará** el cambio y iniciará un nuevo deploy automáticamente
2. **Durante el build**, se ejecutarán automáticamente:
  - `prisma db push` → Crea las tablas en tu base de datos Neon
  - `prisma generate` → Genera el cliente de Prisma
  - `next build` → Construye la aplicación
3. **Tu aplicación funcionará** correctamente con la base de datos

## VARIABLES DE ENTORNO

Asegúrate de que en tu proyecto de Vercel tengas configuradas:

- `DATABASE_URL` → Tu conexión a Neon.tech
- `NEXTAUTH_URL` → Tu dominio de Vercel
- `NEXTAUTH_SECRET` → Generado automáticamente por Vercel

## RESULTADO ESPERADO

-  La aplicación se despliega sin errores
-  Las tablas se crean automáticamente en la base de datos
-  Los formularios funcionan correctamente
-  Se pueden guardar y recuperar datos

## SOPORTE

Si tienes algún problema después del despliegue, verifica:

1. Los logs de build en Vercel
2. Los logs de runtime en Vercel
3. Que las variables de entorno estén configuradas correctamente