

# Guía de Despliegue en V2Networks.cl

---

## Requisitos Previos

---

- Plan de hosting en V2Networks (Crecimiento, Emprendedor o Negocios)
  - Acceso a cPanel
  - Base de datos MySQL creada en cPanel
- 

## Paso 1: Crear Base de Datos MySQL en cPanel

---

1. **Acceder a cPanel** → Ir a “Bases de datos MySQL”
2. **Crear nueva base de datos:**
  - Nombre: `escuelas_deportivas`
3. **Crear usuario MySQL:**
  - Usuario: `escuelas_app`
  - Contraseña: (usar una contraseña segura)
4. **Asignar usuario a la base de datos:**
  - Seleccionar “**TODOS LOS PRIVILEGIOS**”
  - Click en “**Hacer cambios**”

**Nota:** cPanel añade un prefijo a los nombres. Por ejemplo:

- Base de datos: `tucuenta_escuelas_deportivas`
  - Usuario: `tucuenta_escuelas_app`
- 

## Paso 2: Preparar Archivos para Subir

---

### 2.1 Descargar el proyecto

Descarga los archivos desde DeepAgent usando el botón “Files” en la interfaz.

### 2.2 Copiar el schema MySQL

```
cd nextjs_space
cp prisma/schema.mysql.prisma prisma/schema.prisma
```

### 2.3 Crear archivo de inicio para Passenger

Crea un archivo `app.js` en la raíz de `nextjs_space/` :

```
// app.js - Archivo de inicio para Passenger en cPanel
const { createServer } = require('http');
const { parse } = require('url');
const next = require('next');

const dev = false;
const hostname = '0.0.0.0';
const port = process.env.PORT || 3000;

const app = next({ dev, hostname, port });
const handle = app.getRequestHandler();

app.prepare().then(() => {
  createServer(async (req, res) => {
    try {
      const parsedUrl = parse(req.url, true);
      await handle(req, res, parsedUrl);
    } catch (err) {
      console.error('Error occurred handling', req.url, err);
      res.statusCode = 500;
      res.end('internal server error');
    }
  }).listen(port, (err) => {
    if (err) throw err;
    console.log(`> Ready on http://${hostname}:${port}`);
  });
});
```

## 2.4 Crear archivo .env para producción

Crea/edita el archivo `.env` :

```
# Conexión MySQL - Ajusta con tus datos de cPanel
DATABASE_URL="mysql://tucuenta_escuelas_app:TU_PASSWORD@localhost:3306/tucuenta_escuelas_deportivas"

# NextAuth - Cambia el secret y la URL
NEXTAUTH_SECRET="genera_un_secret_seguro_de_32_caracteres"
NEXTAUTH_URL="https://tudominio.cl"

# AWS S3 (si usas almacenamiento de archivos)
AWS_PROFILE=hosted_storage
AWS_REGION=us-west-2
AWS_BUCKET_NAME=tu-bucket
AWS_FOLDER_PREFIX=escuelas/
```

## 2.5 Modificar next.config.js

Asegúrate de tener esta configuración:

```
/** @type {import('next').NextConfig} */
const nextConfig = {
  output: 'standalone',
  // ... otras configuraciones
};

module.exports = nextConfig;
```

## Paso 3: Build Local (Opcional pero Recomendado)

Si tienes Node.js localmente, haz el build antes de subir:

```
cd nextjs_space
npm install
npx prisma generate
npm run build
```

## Paso 4: Subir Archivos a V2Networks

### Opción A: Usando File Manager de cPanel

1. **Acceder a cPanel** → “Administrador de archivos”
2. **Crear directorio** para la app (ej: `escuelas_app`)
3. **Subir archivos** comprimidos (.zip)
4. **Descomprimir** en el servidor

### Opción B: Usando FTP/SFTP

1. Conectar con FileZilla u otro cliente FTP
2. Subir la carpeta `nextjs_space` al directorio deseado

### Archivos a subir:

```
escuelas_app/
├── app.js           Archivo de inicio
├── .env            Variables de entorno
├── package.json
├── package-lock.json
├── next.config.js
├── prisma/
│   └── schema.prisma (copia de schema.mysql.prisma)
├── app/
├── components/
├── lib/
├── public/
└── ... (resto de archivos)
```

### NO subir:

- `node_modules/`
- `.next/`
- `.build/`

## Paso 5: Configurar Node.js en cPanel

1. **Acceder a cPanel**
2. **Ir a “Software”** → “Setup Node.js App”
3. **Click en “Create Application”**

#### 4. Configurar:

- **Node.js version:** Seleccionar 18.x o superior
- **Application mode:** Production
- **Application root:** `escuelas_app` (o el directorio que creaste)
- **Application URL:** Selecciona tu dominio
- **Application startup file:** `app.js`

#### 5. Click en “Create”

---

## Paso 6: Instalar Dependencias via Terminal

---

1. En **cPanel** → “Setup Node.js App” → Tu aplicación
2. **Copiar el comando “source”** que aparece arriba
3. **Ir a “Terminal”** en cPanel (o conectar vía SSH)
4. **Ejecutar:**

```
# Pegar el comando source copiado, ejemplo:
source /home/tucuenta/nodeenv/escuelas_app/18/bin/activate && cd /home/tucuenta/es-
cuelas_app

# Instalar dependencias
npm install

# Generar cliente Prisma
npx prisma generate

# Crear tablas en MySQL
npx prisma db push

# Ejecutar seed (datos iniciales)
npx prisma db seed

# Build de Next.js
npm run build
```

## Paso 7: Iniciar la Aplicación

---

1. En **cPanel** → “Setup Node.js App”
2. **Seleccionar tu aplicación**
3. **Click en “Restart”**

La aplicación debería estar disponible en tu dominio.

---

## Solución de Problemas

---

### Error: “Cannot find module”

```
# En Terminal con el entorno activado:  
npm install  
npx prisma generate
```

### Error de conexión a MySQL

Verifica que:

- El nombre de la base de datos incluye el prefijo de cPanel
- El usuario y contraseña son correctos
- El usuario tiene permisos sobre la base de datos

### Ver logs de errores

```
# En Terminal:  
cd /home/tucuenta/escuelas_app  
cat logs/error.log  
# o  
tail -f logs/error.log
```

### Reiniciar la aplicación manualmente

```
touch tmp/restart.txt
```

### Error “EACCES permission denied”

```
chmod -R 755 /home/tucuenta/escuelas_app
```

---

## Configuración de Variables de Entorno en cPanel

---

Si prefieres no usar archivo .env, puedes configurar variables directamente:

1. En “Setup Node.js App” → Tu aplicación → “Edit”
2. Sección “Environment variables”
3. Agregar cada variable:
  - DATABASE\_URL = mysql://...
  - NEXTAUTH\_SECRET = tu\_secret
  - NEXTAUTH\_URL = https://tudominio.cl

---

## Actualizar la Aplicación

---

Cuando necesites actualizar:

1. Subir los archivos modificados (excepto node\_modules)

## 2. Conectar al Terminal:

```
source /home/tucuenta/nodeenv/escuelas_app/18/bin/activate && cd /home/tucuenta/es-  
cuelas_app  
npm install  
npx prisma generate  
npm run build  
touch tmp/restart.txt
```

---

## Contacto Soporte V2Networks

Si tienes problemas con la configuración del servidor:

- Chat 24/7 en [v2networks.cl](https://v2networks.cl)
- Sistema de tickets en cPanel
- Soporte telefónico (según tu plan)

---

## Checklist Final

- ☐ Base de datos MySQL creada en cPanel
- ☐ Usuario MySQL con permisos asignados
- ☐ Archivos subidos al servidor
- ☐ schema.mysql.prisma copiado como schema.prisma
- ☐ Aplicación Node.js configurada en cPanel
- ☐ Dependencias instaladas (npm install)
- ☐ Cliente Prisma generado (npx prisma generate)
- ☐ Tablas creadas (npx prisma db push)
- ☐ Datos iniciales cargados (npx prisma db seed)
- ☐ Build ejecutado (npm run build)
- ☐ Aplicación reiniciada
- ☐ Sitio funcionando en tu dominio