Manual de Instalación: Escuelas Deportivas Chile

Ubuntu Server 24.04.03 LTS + PostgreSQL

Versión: 1.0
Fecha: Marzo 2026
Requisitos: Ubuntu Server 24.04.03 LTS con SSH configurado

Índice

---

1. Actualización del Sistema

Conéctese al servidor vía SSH y ejecute:

```
# Actualizar lista de paquetes
sudo apt update

# Actualizar paquetes instalados
sudo apt upgrade -y

# Instalar herramientas esenciales
sudo apt install -y curl wget git build-essential

# Reiniciar si hay actualizaciones del kernel
sudo reboot
```

Después del reinicio, reconecte vía SSH.

---

2. Instalación de Node.js 20 LTS

```
# Instalar NodeSource repository para Node.js 20
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -

# Instalar Node.js
sudo apt install -y nodejs

# Verificar instalación
node --version   # Debe mostrar v20.x.x
npm --version    # Debe mostrar 10.x.x

# Instalar Yarn globalmente
sudo npm install -g yarn

# Verificar Yarn
yarn --version   # Debe mostrar 1.22.x
```

---

3. Instalación de PostgreSQL 16

```
# Agregar repositorio oficial de PostgreSQL
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg
main" > /etc/apt/sources.list.d/pgdg.list'

# Importar clave GPG del repositorio
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key ad
d -

# Actualizar lista de paquetes
sudo apt update

# Instalar PostgreSQL 16
sudo apt install -y postgresql-16 postgresql-contrib-16

# Verificar que PostgreSQL está corriendo
sudo systemctl status postgresql

# Habilitar inicio automático
sudo systemctl enable postgresql

# Verificar versión
sudo -u postgres psql -c "SELECT version();"
```

---

4. Configuración de PostgreSQL

4.1 Crear Usuario y Base de Datos

```
# Acceder a PostgreSQL como superusuario
sudo -u postgres psql
```

Dentro de la consola de PostgreSQL, ejecute:

```
-- Crear usuario para la aplicación (cambie 'MiPassword123!' por una contraseña se-
gura)
CREATE USER escuelas_user WITH PASSWORD 'MiPassword123!';

-- Crear base de datos
CREATE DATABASE escuelas_deportivas
    WITH OWNER = escuelas_user
    ENCODING = 'UTF8'
    LC_COLLATE = 'es_CL.UTF-8'
    LC_CTYPE = 'es_CL.UTF-8'
    TEMPLATE = template0;

-- Otorgar privilegios completos
GRANT ALL PRIVILEGES ON DATABASE escuelas_deportivas TO escuelas_user;

-- Conectar a la base de datos
\c escuelas_deportivas

-- Otorgar privilegios en el esquema público
GRANT ALL ON SCHEMA public TO escuelas_user;

-- Salir
\q
```

4.2 Configurar Autenticación (Opcional para conexiones remotas)

```
# Editar pg_hba.conf para permitir conexiones
sudo nano /etc/postgresql/16/main/pg_hba.conf
```

Agregar o modificar la línea (para conexiones locales con contraseña):

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD
local   escuelas_deportivas  escuelas_user                        md5
host    escuelas_deportivas  escuelas_user  127.0.0.1/32          md5
host    escuelas_deportivas  escuelas_user  ::1/128               md5
```

```
# Reiniciar PostgreSQL para aplicar cambios
sudo systemctl restart postgresql

# Verificar conexión
psql -h localhost -U escuelas_user -d escuelas_deportivas -c "SELECT cur-
rent_database();"
```

---

5. Preparación del Proyecto

5.1 Crear Usuario de Aplicación (Recomendado)

```
# Crear usuario para la aplicación
sudo useradd -m -s /bin/bash escuelas_app

# Establecer contraseña
sudo passwd escuelas_app

# Cambiar a usuario de aplicación
sudo su - escuelas_app
```

5.2 Crear Directorio y Subir Archivos

Opción A: Subir archivos vía SCP/SFTP

```
# Desde su máquina local:
scp -r ./nextjs_space escuelas_app@IP_SERVIDOR:/home/escuelas_app/escuelas_deportivas/
```

Opción B: Clonar desde repositorio Git

```
cd /home/escuelas_app
git clone https://su-repositorio.git escuelas_deportivas
```

Opción C: Descomprimir archivo

```
cd /home/escuelas_app
mkdir escuelas_deportivas
tar -xzf escuelas_deportivas.tar.gz -C escuelas_deportivas
```

5.3 Estructura Esperada

```
/home/escuelas_app/escuelas_deportivas/
├── app/
│   ├── api/
│   ├── globals.css
│   ├── layout.tsx
│   └── page.tsx
├── components/
├── lib/
├── prisma/
│   └── schema.prisma
├── scripts/
│   └── seed.ts
├── public/
├── next.config.js
├── package.json
├── tsconfig.json
└── .env (crear)
```

6. Configuración de Variables de Entorno

```
cd /home/escuelas_app/escuelas_deportivas

# Copiar archivo de ejemplo
cp .env.postgresql.example .env

# Editar con sus valores
nano .env
```

El proyecto incluye un archivo de ejemplo `.env.postgresql.example`. Contenido del archivo `.env`:

```
# ============================================
# CONFIGURACIÓN DE BASE DE DATOS POSTGRESQL
# ============================================
# Formato: postgresql://USUARIO:PASSWORD@HOST:PUERTO/BASE_DE_DATOS?schema=public
DATABASE_URL="postgresql://escuelas_user:MiPassword123!@localhost:5432/
escuelas_deportivas?schema=public"

# ============================================
# CONFIGURACIÓN DE NEXTAUTH
# ============================================
# Generar con: openssl rand -base64 32
NEXTAUTH_SECRET="generar-una-clave-secreta-de-32-caracteres"

# URL de producción (cambiar por su dominio)
NEXTAUTH_URL="https://escuelasdeportivas.cl"

# ============================================
# CONFIGURACIÓN DE AWS S3 (Opcional)
# ============================================
# Solo necesario si usa almacenamiento de archivos
#AWS_ACCESS_KEY_ID=""
#AWS_SECRET_ACCESS_KEY=""
#AWS_REGION="us-east-1"
#AWS_BUCKET_NAME=""
#AWS_FOLDER_PREFIX="escuelas/"

# ============================================
# ENTORNO
# ============================================
NODE_ENV="production"
```

Generar NEXTAUTH_SECRET

```
openssl rand -base64 32
```

Copie el resultado y reemplace `generar-una-clave-secreta-de-32-caracteres` en el archivo `.env`.

---

7. Instalación de Dependencias y Base de Datos

```
cd /home/escuelas_app/escuelas_deportivas

# Instalar dependencias de Node.js
yarn install

# Verificar que Prisma está instalado
yarn prisma --version

# Generar cliente de Prisma
yarn prisma generate

# Crear tablas en la base de datos
yarn prisma db push

# Ejecutar script de datos iniciales (seed)
yarn prisma db seed
```

Verificar Tablas Creadas

```
psql -h localhost -U escuelas_user -d escuelas_deportivas -c "\dt"
```

Debería ver las tablas: `User`, `Deporte`, `Provincia`, `Comuna`, `Escuela`, `Instructor`, etc.

---

8. Compilación de la Aplicación

```
cd /home/escuelas_app/escuelas_deportivas

# Compilar para producción
yarn build

# Probar ejecución local (Ctrl+C para detener)
yarn start
```

Si funciona correctamente, verá:

```
   ▲ Next.js 14.x.x
   - Local:        http://localhost:3000
```

---

9. Configuración de PM2 (Gestor de Procesos)

9.1 Instalar PM2

```
# Instalar PM2 globalmente
sudo npm install -g pm2

# Verificar instalación
pm2 --version
```

9.2 Crear Archivo de Configuración PM2

```
cd /home/escuelas_app/escuelas_deportivas
nano ecosystem.config.js
```

Contenido:

```
module.exports = {
  apps: [{
    name: 'escuelas-deportivas',
    script: 'node_modules/.bin/next',
    args: 'start',
    cwd: '/home/escuelas_app/escuelas_deportivas',
    instances: 'max',
    exec_mode: 'cluster',
    env: {
      NODE_ENV: 'production',
      PORT: 3000
    },
    max_memory_restart: '500M',
    error_file: '/home/escuelas_app/logs/err.log',
    out_file: '/home/escuelas_app/logs/out.log',
    merge_logs: true,
    time: true
  }]
};
```

9.3 Crear Directorio de Logs

```
mkdir -p /home/escuelas_app/logs
```

9.4 Iniciar Aplicación

```
# Iniciar con PM2
pm2 start ecosystem.config.js

# Verificar estado
pm2 status

# Ver logs en tiempo real
pm2 logs escuelas-deportivas

# Configurar inicio automático
pm2 startup systemd
# Ejecutar el comando que PM2 muestra

# Guardar configuración actual
pm2 save
```

---

10. Configuración de Nginx (Proxy Reverso)

10.1 Instalar Nginx

```
sudo apt install -y nginx

# Verificar instalación
sudo systemctl status nginx

# Habilitar inicio automático
sudo systemctl enable nginx
```

10.2 Configurar Virtual Host

```
sudo nano /etc/nginx/sites-available/escuelas-deportivas
```

Contenido:

```
server {
    listen 80;
    server_name escuelasdeportivas.cl www.escuelasdeportivas.cl;

    # Logs
    access_log /var/log/nginx/escuelas_access.log;
    error_log /var/log/nginx/escuelas_error.log;

    # Proxy hacia Next.js
    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
        proxy_read_timeout 90s;
        proxy_connect_timeout 90s;
    }

    # Archivos estáticos de Next.js
    location /_next/static {
        proxy_pass http://localhost:3000;
        proxy_cache_valid 60m;
        add_header Cache-Control "public, max-age=31536000, immutable";
    }

    # Archivos públicos
    location /public {
        alias /home/escuelas_app/escuelas_deportivas/public;
        expires 30d;
        add_header Cache-Control "public, max-age=2592000";
    }

    # Tamaño máximo de upload
    client_max_body_size 10M;
}
```

10.3 Habilitar Sitio

```
# Crear enlace simbólico
sudo ln -s /etc/nginx/sites-available/escuelas-deportivas /etc/nginx/sites-enabled/

# Verificar configuración
sudo nginx -t

# Recargar Nginx
sudo systemctl reload nginx
```

---

11. Certificado SSL con Let’s Encrypt

11.1 Instalar Certbot

```
sudo apt install -y certbot python3-certbot-nginx
```

11.2 Obtener Certificado

```
# Asegúrese de que su dominio apunta a la IP del servidor
sudo certbot --nginx -d escuelasdeportivas.cl -d www.escuelasdeportivas.cl
```

Siga las instrucciones interactivas. Certbot:
- Verificará la propiedad del dominio
- Obtendrá el certificado
- Configurará Nginx automáticamente
- Configurará renovación automática

11.3 Verificar Renovación Automática

```
sudo certbot renew --dry-run
```

---

12. Firewall y Seguridad

12.1 Configurar UFW

```
# Habilitar UFW
sudo ufw enable

# Permitir SSH
sudo ufw allow ssh

# Permitir HTTP y HTTPS
sudo ufw allow 'Nginx Full'

# Verificar reglas
sudo ufw status verbose
```

12.2 Seguridad de PostgreSQL

Por defecto, PostgreSQL solo acepta conexiones locales. Para mayor seguridad:

```
# Verificar que PostgreSQL escucha solo en localhost
sudo grep listen_addresses /etc/postgresql/16/main/postgresql.conf
# Debe mostrar: listen_addresses = 'localhost'
```

---

13. Verificación Final

Lista de Verificación

```
# 1. PostgreSQL corriendo
sudo systemctl status postgresql

# 2. PM2 corriendo
pm2 status

# 3. Nginx corriendo
sudo systemctl status nginx

# 4. Aplicación respondiendo
curl -I http://localhost:3000

# 5. Nginx respondiendo
curl -I http://localhost

# 6. Verificar logs de errores
pm2 logs escuelas-deportivas --lines 50
sudo tail -50 /var/log/nginx/escuelas_error.log
```

Credenciales por Defecto

- **Usuario Admin:** admin@escuelasdeportivas.cl
- **Contraseña:** admin123

⚠️ **IMPORTANTE:** Cambie la contraseña del administrador después del primer inicio de sesión.

---

14. Comandos Útiles

PM2

```
# Ver estado
pm2 status

# Reiniciar aplicación
pm2 restart escuelas-deportivas

# Detener aplicación
pm2 stop escuelas-deportivas

# Ver logs
pm2 logs escuelas-deportivas

# Monitoreo en tiempo real
pm2 monit
```

Prisma

```
# Abrir Prisma Studio (interfaz visual de BD)
yarn prisma studio

# Sincronizar esquema
yarn prisma db push

# Regenerar cliente
yarn prisma generate

# Re-ejecutar seed
yarn prisma db seed
```

PostgreSQL

```
# Conectar a la base de datos
psql -h localhost -U escuelas_user -d escuelas_deportivas

# Ver tablas
\dt

# Ver contenido de una tabla
SELECT * FROM "User" LIMIT 5;

# Salir
\q
```

Nginx

```
# Verificar configuración
sudo nginx -t

# Recargar
sudo systemctl reload nginx

# Reiniciar
sudo systemctl restart nginx

# Ver logs
sudo tail -f /var/log/nginx/escuelas_access.log
```

---

15. Solución de Problemas

Error: “EACCES: permission denied”

```
# Verificar propietario de archivos
ls -la /home/escuelas_app/escuelas_deportivas

# Corregir permisos
sudo chown -R escuelas_app:escuelas_app /home/escuelas_app/escuelas_deportivas
chmod -R 755 /home/escuelas_app/escuelas_deportivas
```

Error: “Connection refused” (PostgreSQL)

```
# Verificar que PostgreSQL está corriendo
sudo systemctl status postgresql

# Verificar configuración de pg_hba.conf
sudo cat /etc/postgresql/16/main/pg_hba.conf | grep escuelas

# Reiniciar PostgreSQL
sudo systemctl restart postgresql
```

Error: “Database does not exist”

```
# Verificar bases de datos existentes
sudo -u postgres psql -l

# Crear base de datos si no existe
sudo -u postgres createdb escuelas_deportivas -O escuelas_user
```

Error: “Prisma Client not generated”

```
cd /home/escuelas_app/escuelas_deportivas
yarn prisma generate
```

Error de Memoria (Out of Memory)

```
# Crear archivo swap (2GB)
sudo fallocate -l 2G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile

# Hacer permanente
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
```

La aplicación no inicia después de reiniciar el servidor

```
# Verificar que PM2 está configurado para inicio automático
pm2 startup
pm2 save
```

Error 502 Bad Gateway en Nginx

```
# Verificar que la aplicación está corriendo
pm2 status

# Si no está corriendo, iniciar
pm2 start ecosystem.config.js

# Verificar logs de Nginx
sudo tail -50 /var/log/nginx/escuelas_error.log
```

Actualizar la Aplicación

```
cd /home/escuelas_app/escuelas_deportivas

# Detener aplicación
pm2 stop escuelas-deportivas

# Actualizar archivos (según método de deploy)
git pull origin main  # Si usa Git
# o subir nuevos archivos vía SCP/SFTP

# Instalar nuevas dependencias
yarn install

# Aplicar cambios de esquema
yarn prisma generate
yarn prisma db push

# Recompilar
yarn build

# Reiniciar
pm2 restart escuelas-deportivas
```

Resumen de Puertos

Puerto	Servicio	Acceso
22	SSH	Público
80	HTTP (Nginx)	Público
443	HTTPS (Nginx)	Público
3000	Next.js	Solo Local
5432	PostgreSQL	Solo Local

Soporte

Para problemas técnicos:
1. Revisar logs de PM2: `pm2 logs escuelas-deportivas`
2. Revisar logs de Nginx: `sudo tail -f /var/log/nginx/escuelas_error.log`
3. Revisar logs de PostgreSQL: `sudo tail -f /var/log/postgresql/postgresql-16-main.log`

Documento generado para Escuelas Deportivas Chile
Ubuntu Server 24.04.03 LTS + PostgreSQL 16