

Manual de Despliegue para Producción

Francisco Damián Méndez Palma
IES Francisco de Orellana - Trujillo

Sumario

Estructura básica del proyecto.....	3
1. Instalaciones previas.....	3
2. Configurar variables de entorno.....	3
3. Despliegue del Backend con PM2.....	4
4. Construcción del Frontend (Vite + ShadCN).....	4
5. Configurar Nginx.....	4
7. Generar certificados autofirmados para HTTPS.....	6

Estructura básica del proyecto

```
/gestionIES
├── backend/      # Backend Node.js
│   ├── controllers/
│   ├── routes/
│   ├── server.js
│   └── ...
├── src/          # Frontend React + Vite + ShadCN
│   ├── index.html
│   └── ...
├── .env
└── ...
```

1. Instalaciones previas

Instalar:

```
sudo apt update
sudo apt install nodejs npm nginx postgresql -y
```

También instalar **PM2** (para gestionar el backend):

```
sudo npm install -g pm2
```

2. Configurar variables de entorno

Editar `.env` de `gestionIES/backend`. Ahora estamos en modo producción, ya que vamos a desplegar la aplicación en el servidor que la alojará.

```
# Modo desarrollo
#NODE_ENV=development
# Modo producción - al desplegar en servidor
NODE_ENV=production

# Puerto backend - node.js (mismo que en nginx)
PORT=5000
SESSION_SECRET=clave-sesion-segura
# Orígenes permitidos para acceso CORS
ALLOWED_ORIGINS=http://localhost:5173,https://localhost:5173,https://<tu_ip>:5173

#
# Configuración base de datos PostgreSQL
#

# IES desarrollo
#DB_HOST=<tu_ip>

# IES produccion
DB_HOST=<ip_produccion>

DB_PORT=5432
DB_USER=postgres
DB_PASSWORD=<password postgres>
```

```
DB_NAME=<nombre_tu_bd>
```

```
#  
# Configuración servidor LDAP  
#
```

```
# IES Desarrollo y produccion (ya que de momento solo hace bind para lecturas)  
LDAP_URL=ldap://<ip_ldap>:389
```

3. Despliegue del Backend con PM2

Desde la carpeta gestionIES/backend/:

```
npm install  
pm2 start server.js --name backend  
pm2 save
```

4. Construcción del Frontend (Vite + ShadCN)

Desde gestionIES/src:

```
npm install  
npm run build
```

Esto generará una carpeta dist/.

5. Configurar Nginx

Abrir y editar el archivo de configuración:

```
sudo nano /etc/nginx/sites-available/gestionIES
```

Contenido:

```
server {  
    listen 80;  
    server_name 172.16.218.200;  
    return 301 https://$host$request_uri;  
}
```

```
server {  
    listen 443 ssl;  
    server_name <ip_servidor_que_aloja_la_app>;  
  
    ssl_certificate /etc/nginx/ssl/nginx.crt;  
    ssl_certificate_key /etc/nginx/ssl/nginx.key;  
    ssl_session_timeout 1d;  
    ssl_session_cache shared:SSL:10m;  
    ssl_session_tickets off;  
    ssl_protocols TLSv1.2 TLSv1.3;  
    ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-  
ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-  
SHA256:DHE-RSA-AES256-GCM-SHA384';  
    ssl_prefer_server_ciphers off;
```

```
# No hay 'root' o 'index' globales aquí.

# Bloque principal para servir el frontend SPA
location / {
    # Usa 'alias' para el directorio de los archivos estáticos.
    # Asegúrate de que termina con un slash si el 'location' termina con un slash.
    alias /var/www/gestionIES/dist;

    # Intenta servir el URI tal cual (e.g., /assets/index-xxxxxxx.js)
    # Si no lo encuentra, usa el index.html como fallback.
    try_files $uri $uri/index.html $uri/ =404;

    # Si 'try_files' no encuentra nada, aún necesitamos que el index.html se sirva.
    # Esta es la parte que a veces ayuda a romper el ciclo.
    # Nginx no suele hacer un redirect con 'try_files' si el último elemento es un archivo.
    # El problema es que si $uri/index.html es el fallback y se genera un ciclo,
    # es porque Nginx vuelve a evaluar la petición.

    # Vamos a intentar con un 'error_page' para capturar el 404 y servir index.html
    error_page 404 = /index.html;
}
```

FUNCIONA EN PRODUCCION

```
location /api/ {
    proxy_pass http://localhost:5000/api/;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;

    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_cookie_path / "/; SameSite=Lax; Secure; HttpOnly";
}
```

Servir imágenes o archivos estáticos de /public (si no están ya en el frontend build)

```
location /public/ {
    alias /var/www/gestionIES/public/;
    expires 30d;
    add_header Cache-Control "public";
}
```

servir imagenes de alumnos

```
location /uploads/ {
    alias /var/www/gestionIES/backend/uploads/;
    expires 30d;
    add_header Cache-Control "public";
}
}
```

Activar el sitio:

```
sudo ln -s /etc/nginx/sites-available/mi-app /etc/nginx/sites-enabled/  
sudo nginx -t  
sudo systemctl restart nginx
```

7. Generar certificados autofirmados para HTTPS.

Hay que generar los certificados autofirmados para acceder por https a la aplicación.

```
sudo mkdir -p /etc/nginx/ssl  
cd /etc/nginx/ssl  
  
# Generar clave privada y certificado autofirmado, válidos por 365 días  
sudo openssl req -x509 -nodes -days 365 \  
-newkey rsa:2048 \  
-keyout nginx.key \  
-out nginx.crt
```

Durante el proceso, openssl pedirá una serie de datos (Country, State, Common Name, etc.). Puedes dejarlos vacíos o introducir los que consideres.

De esta forma, cualquier navegador en la red interna podrá acceder por https://ip_servidor_aplicacion, aunque el navegador avisará de que no es “seguro” porque es autofirmado (no está validado por una CA).

8. Subir cambios del código a producción

Para subir tus cambios de desarrollo a producción, hay que seguir los siguientes pasos:

- Copiar el código de tu equipo de desarrollo a producción (/var/www/gestionIES). Puede hacerse simplemente con un `scp -r` o con un `rsync`.
- Actualizar dependencias en backend y frontend.
 - Ir a `/var/www/gestionIES` y ejecutar `npm install`
 - Ir a `/var/www/gestionIES/backend` y ejecutar `npm install`.
- Construir el frontend:
 - `npm run build`.
- Reiniciar backend con `pm2 restart`.
- Recargar nginx con `systemctl reload nginx`.