# 🚀 GUÍA COMPLETA DE INSTALACIÓN EN DIGITALOCEAN UBUNTU - VERSIÓN MEJORADA CON MYSQL

## 📋 INFORMACIÓN DEL SERVIDOR

* **Dominio Frontend:**csdt.geekcorplab.com
* **Dominio API:**api-csdt.geekcorplab.com
* **IP Pública:**64.225.113.49
* **IP Privada:**10.120.0.2
* **Repositorio Frontend:**https://github.com/veeduriacsdc-ui/frontend-csdt
* **Repositorio Backend:**https://github.com/veeduriacsdc-ui/backend-csdt.git
* **Base de Datos:** MySQL 8.0 (Producción)

## 🔄 DIFERENCIA ENTRE XAMPP (DESARROLLO) Y MYSQL (PRODUCCIÓN)

### **XAMPP (Desarrollo Local):**

* ✅ **Uso:** Solo para desarrollo local en tu computadora
* ✅ **Base de datos:** SQLite (archivo local) o MySQL (XAMPP)
* ✅ **Configuración:**.env con DB\_CONNECTION=sqlite o DB\_CONNECTION=mysql
* ✅ **Ventajas:** Fácil de instalar, no requiere configuración de servidor
* ❌ **Limitaciones:** No es adecuado para producción, no maneja múltiples usuarios

### **MySQL 8.0 (Producción en DigitalOcean):**

* ✅ **Uso:** Para servidores de producción (DigitalOcean, AWS, etc.)
* ✅ **Base de datos:** MySQL 8.0 (servidor dedicado)
* ✅ **Configuración:**.env con DB\_CONNECTION=mysql
* ✅ **Ventajas:** Robusto, escalable, maneja múltiples usuarios, optimizado para producción
* ✅ **Seguridad:** Configuración segura, usuarios específicos, permisos granulares

### **¿Por qué MySQL en producción?**

1. **Rendimiento:** MySQL es mucho más rápido que SQLite para múltiples usuarios
2. **Concurrencia:** Maneja múltiples usuarios simultáneos sin problemas
3. **Seguridad:** Usuarios y permisos específicos para cada aplicación
4. **Escalabilidad:** Puede crecer con tu aplicación
5. **Backup:** Herramientas profesionales de backup y recuperación
6. **Monitoreo:** Herramientas avanzadas de monitoreo y optimización

## 🔧 PASO 1: CONECTAR Y PREPARAR EL SERVIDOR

### 1.1: Conectar al servidor

ssh root@64.225.113.49

### 1.2: Actualizar sistema

sudo apt update && sudo apt upgrade -y

### 1.3: Instalar dependencias básicas

# Instalar PHP 8.2 y extensiones necesariassudo apt install software-properties-common -y sudo add-apt-repository ppa:ondrej/php -y sudo apt update sudo apt install php8.2 php8.2-cli php8.2-fpm php8.2-mysql php8.2-xml php8.2-mbstring php8.2-curl php8.2-zip php8.2-bcmath php8.2-gd php8.2-sqlite3 -y # Instalar Composer curl -sS https://getcomposer.org/installer | php sudomv composer.phar /usr/local/bin/composer sudochmod +x /usr/local/bin/composer # Instalar Node.js 18.x curl -fsSL https://deb.nodesource.com/setup\_18.x | sudo -E bash - sudo apt-get install -y nodejs # Instalar Gitsudo apt install git -y # Instalar Nginxsudo apt install nginx -y # Instalar PM2sudo npm install -g pm2 # Instalar MySQL 8.0 (OBLIGATORIO para producción)sudo apt install mysql-server-8.0 -y # Instalar herramientas adicionalessudo apt install unzip wget curl -y # Verificar instalaciones php --version composer --version node --version npm --version git --version sudo nginx -v sudo pm2 --version

## 📁 PASO 2: CONFIGURAR EL BACKEND (LARAVEL)

### 2.1: Crear directorio del backend

# Crear directoriosudomkdir -p /var/www/csdt-backend cd /var/www/csdt-backend # Verificar ubicaciónpwd

### 2.2: Clonar repositorio del backend

# Clonar repositoriosudo git clone https://github.com/veeduriacsdc-ui/backend-csdt.git . # Verificar que se clonóls -la

### 2.3: Instalar dependencias del backend

# Instalar dependencias PHPsudo composer install --optimize-autoloader --no-dev # Instalar dependencias Node.js (solo para Vite, NO ejecutar build aquí)sudo npm install # Verificar instalaciónls -la vendor/ ls -la node\_modules/

### 2.4: Crear archivo .env para el backend

# Crear archivo .envsudo nano .env

**Contenido del archivo .env:**

# =========================================== # CONFIGURACIÓN DE LA APLICACIÓN # =========================================== APP\_NAME="CONSEJO SOCIAL DE VEEDURÍA Y DESARROLLO TERRITORIAL" APP\_ENV=production APP\_KEY= APP\_DEBUG=false APP\_URL=https://api-csdt.geekcorplab.com APP\_TIMEZONE=America/Bogota APP\_LOCALE=es APP\_FALLBACK\_LOCALE=es # =========================================== # CONFIGURACIÓN DE BASE DE DATOS (MYSQL) # =========================================== # IMPORTANTE: En producción (DigitalOcean) usamos MySQL 8.0 # NO usar SQLite ni XAMPP en servidores de producción DB\_CONNECTION=mysql DB\_HOST=127.0.0.1 DB\_PORT=3306 DB\_DATABASE=csdt\_final DB\_USERNAME=csdt DB\_PASSWORD=tu\_password\_seguro\_aqui DB\_CHARSET=utf8mb4 DB\_COLLATION=utf8mb4\_unicode\_ci # =========================================== # CONFIGURACIÓN PARA DESARROLLO LOCAL (XAMPP) # =========================================== # NOTA: Solo usar estas configuraciones en tu computadora local con XAMPP # Para SQLite (desarrollo local): # DB\_CONNECTION=sqlite # DB\_DATABASE=/var/www/csdt-backend/database/database.sqlite # Para MySQL con XAMPP (desarrollo local): # DB\_CONNECTION=mysql # DB\_HOST=127.0.0.1 # DB\_PORT=3306 # DB\_DATABASE=csdt\_final # DB\_USERNAME=root # DB\_PASSWORD= # =========================================== # CONFIGURACIÓN DE CACHE Y SESIONES # =========================================== CACHE\_DRIVER=file SESSION\_DRIVER=file SESSION\_LIFETIME=120 QUEUE\_CONNECTION=sync # =========================================== # CONFIGURACIÓN DE MAIL # =========================================== MAIL\_MAILER=smtp MAIL\_HOST=smtp.gmail.com MAIL\_PORT=587 MAIL\_USERNAME=tu\_email@gmail.com MAIL\_PASSWORD=tu\_password\_de\_aplicacion MAIL\_ENCRYPTION=tls MAIL\_FROM\_ADDRESS=noreply@csdt.geekcorplab.com MAIL\_FROM\_NAME="${APP\_NAME}" # =========================================== # CONFIGURACIÓN DE SANCTUM # =========================================== SANCTUM\_STATEFUL\_DOMAINS=csdt.geekcorplab.com,api-csdt.geekcorplab.com SESSION\_DOMAIN=.geekcorplab.com # =========================================== # CONFIGURACIÓN DE CORS # =========================================== CORS\_ALLOWED\_ORIGINS=https://csdt.geekcorplab.com CORS\_ALLOWED\_HEADERS=\* CORS\_ALLOWED\_METHODS=GET,POST,PUT,DELETE,OPTIONS CORS\_SUPPORTS\_CREDENTIALS=true # =========================================== # CONFIGURACIÓN DE LOGS # =========================================== LOG\_CHANNEL=stack LOG\_LEVEL=info LOG\_DEPRECATIONS\_CHANNEL=null # =========================================== # CONFIGURACIÓN DE ARCHIVOS # =========================================== FILESYSTEM\_DISK=local UPLOAD\_PATH=/var/www/csdt-backend/storage/uploads MAX\_FILE\_SIZE=10485760 ALLOWED\_FILE\_TYPES=jpg,jpeg,png,pdf,doc,docx # =========================================== # CONFIGURACIÓN DE SENTRY (OPCIONAL) # =========================================== SENTRY\_LARAVEL\_DSN= SENTRY\_TRACES\_SAMPLE\_RATE=0.1 # =========================================== # CONFIGURACIÓN DE DESARROLLO # =========================================== VITE\_APP\_NAME="${APP\_NAME}" VITE\_APP\_ENV=production

### 2.5: Configurar MySQL

# Configurar MySQL de forma segurasudo mysql\_secure\_installation # Crear base de datos y usuariosudo mysql -u root -p

**Dentro de MySQL, ejecutar:**

-- Crear base de datosCREATE DATABASE csdt\_final CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci; -- Crear usuarioCREATEUSER'csdt'@'localhost' IDENTIFIED BY'123'; -- Otorgar permisosGRANTALL PRIVILEGES ON csdt\_final.\*TO'csdt'@'localhost'; -- Aplicar cambios FLUSH PRIVILEGES; -- VerificarSHOW DATABASES; SELECTUser, Host FROM mysql.user WHEREUser='csdt'; -- Salir EXIT;

### 2.6: Configurar la aplicación Laravel

# Generar clave de aplicaciónsudo php artisan key:generate # Verificar conexión a MySQLsudo php artisan tinker # Dentro de tinker: DB::connection()->getPdo();# Salir: exit# Ejecutar migracionessudo php artisan migrate --force # Ejecutar seederssudo php artisan db:seed --force # Limpiar y optimizar cachesudo php artisan config:cache sudo php artisan route:cache sudo php artisan view:cache # IMPORTANTE: NO ejecutar npm run build aquí (solo en frontend)# El backend usa Vite para compilar assets de Inertia.js# Configurar permisossudochown -R www-data:www-data /var/www/csdt-backend sudochmod -R 755 /var/www/csdt-backend sudochmod -R 775 /var/www/csdt-backend/storage sudochmod -R 775 /var/www/csdt-backend/bootstrap/cache

### 2.7: Crear archivo de configuración de PM2

# Crear archivo de configuraciónsudo nano ecosystem.config.js

**Contenido del archivo ecosystem.config.js:**

module.exports = { apps: [{ name: 'csdt-backend', script: 'artisan', args: 'serve --host=0.0.0.0 --port=8000', instances: 1, exec\_mode: 'fork', env: { NODE\_ENV: 'development', APP\_ENV: 'local' }, env\_production: { NODE\_ENV: 'production', APP\_ENV: 'production' }, log\_file: '/var/log/csdt-backend/combined.log', out\_file: '/var/log/csdt-backend/out.log', error\_file: '/var/log/csdt-backend/error.log', log\_date\_format: 'YYYY-MM-DD HH:mm:ss Z', merge\_logs: true, max\_memory\_restart: '512M', watch: false, ignore\_watch: ['node\_modules', 'storage/logs', 'vendor'], restart\_delay: 4000, max\_restarts: 10, min\_uptime: '10s' }] };

### 2.8: Configurar PM2 para el backend

# Crear directorio de logssudomkdir -p /var/log/csdt-backend sudochown -R www-data:www-data /var/log/csdt-backend # Iniciar la aplicación con PM2sudo pm2 start ecosystem.config.js --env production # Verificar que esté corriendosudo pm2 status # Configurar para que inicie automáticamentesudo pm2 startup sudo pm2 save

## 🖥️ PASO 3: CONFIGURAR EL FRONTEND (REACT + VITE)

### 3.1: Crear directorio del frontend

# Crear directoriosudomkdir -p /var/www/csdt-frontend cd /var/www/csdt-frontend # Verificar ubicaciónpwd

### 3.2: Clonar repositorio del frontend

# Clonar repositoriosudo git clone https://github.com/veeduriacsdc-ui/frontend-csdt.git . # Verificar que se clonóls -la

### 3.3: Instalar dependencias del frontend

# Instalar dependenciassudo npm install # Verificar instalaciónls -la node\_modules/

### 3.4: Crear archivo .env para el frontend

# Crear archivo .envsudo nano .env

**Contenido del archivo .env:**

# =========================================== # CONFIGURACIÓN DE LA APLICACIÓN # =========================================== VITE\_APP\_NAME="CONSEJO SOCIAL DE VEEDURÍA Y DESARROLLO TERRITORIAL" VITE\_APP\_VERSION="1.0.0" VITE\_APP\_ENV=production # =========================================== # CONFIGURACIÓN DE API # =========================================== VITE\_API\_URL=https://api-csdt.geekcorplab.com VITE\_API\_TIMEOUT=30000 # =========================================== # CONFIGURACIÓN DE SERVICIOS DE IA # =========================================== VITE\_IA\_ENABLED=true VITE\_IA\_SERVICES\_ENABLED=true VITE\_IA\_MEJORADA\_ENABLED=true VITE\_IA\_PROFESIONALES\_ENABLED=true VITE\_IA\_SISTEMA\_PROFESIONAL\_ENABLED=true VITE\_IA\_CHAT\_MEJORADO\_ENABLED=true VITE\_IA\_TECNICAS\_ENABLED=true VITE\_IA\_CONSEJO\_ENABLED=true # =========================================== # CONFIGURACIÓN DE MAPAS # =========================================== VITE\_LEAFLET\_ENABLED=true VITE\_MAPBOX\_TOKEN=tu\_mapbox\_token\_aqui VITE\_GEODJANGO\_ENABLED=true # =========================================== # CONFIGURACIÓN DE NOTIFICACIONES # =========================================== VITE\_NOTIFICATIONS\_ENABLED=true VITE\_PUSH\_NOTIFICATIONS=false # =========================================== # CONFIGURACIÓN DE DESARROLLO # =========================================== VITE\_DEBUG=false VITE\_VERBOSE\_LOGGING=false

### 3.5: Instalar y configurar servicios de IA completos

# Verificar que todos los servicios de IA estén presentesls -la src/services/ # Los siguientes servicios deben estar presentes:# - IAMejoradaService.js# - IAsProfesionalesService.js# - SistemaIAProfesionalService.js# - ChatGPTMejoradoService.js# - IAsTecnicasService.js# - ConsejoIAService.js# - AnalisisJuridicoService.js# - AnalisisNarrativoProfesionalService.js

### 3.5.1: Servicios de IA (se cargan automáticamente con el repositorio)

# NOTA: Los servicios de IA se cargan automáticamente con el repositorio# IAMejoradaService.js incluye análisis especializados para las 7 acciones constitucionalesecho"✅ Servicios de IA cargados automáticamente desde el repositorio"

### 3.5.9: Configurar variables de entorno para IA

# Agregar variables de IA al archivo .env del frontendsudo nano .env

**Agregar al archivo .env:**

# =========================================== # CONFIGURACIÓN DE SERVICIOS DE IA # =========================================== VITE\_OPENAI\_API\_KEY=tu\_api\_key\_de\_openai\_aqui VITE\_IA\_TIMEOUT=30000 VITE\_IA\_MAX\_TOKENS=2000 VITE\_IA\_TEMPERATURE=0.7

### 3.5.10: Verificar servicios de IA (se cargan automáticamente con el repositorio)

# NOTA: Los servicios de IA se cargan automáticamente con el repositorio# Solo verificar que estén presentes:ls -la src/services/ # Los siguientes servicios deben estar presentes automáticamente:# - IAMejoradaService.js (versión mejorada con análisis especializados)# - IAsProfesionalesService.js# - SistemaIAProfesionalService.js# - ChatGPTMejoradoService.js# - IAsTecnicasService.js# - ConsejoIAService.js# - AnalisisJuridicoService.js# - AnalisisNarrativoProfesionalService.jsecho"✅ Servicios de IA cargados automáticamente desde el repositorio"

### 3.6: Compilar el frontend

# Compilar para producción (AQUÍ SÍ se ejecuta el build)sudo npm run build # Verificar compilaciónls -la dist/ # Configurar permisossudochown -R www-data:www-data /var/www/csdt-frontend sudochmod -R 755 /var/www/csdt-frontend

## 🌐 PASO 4: CONFIGURAR NGINX

### 4.1: Crear configuración para el frontend

# Crear archivo de configuraciónsudo nano /etc/nginx/sites-available/csdt-frontend

**Contenido del archivo:**

server { listen80; server\_name csdt.geekcorplab.com; root /var/www/csdt-frontend/dist; index index.html; # Configuración para SPA (Single Page Application)location / { try\_files$uri$uri/ /index.html; } # Configuración para archivos estáticoslocation~\* \.(js|css|png|jpg|jpeg|gif|ico|svg|woff|woff2|ttf|eot)$ { expires1y; add\_header Cache-Control "public, immutable"; try\_files$uri =404; } # Configuración de seguridadadd\_header X-Frame-Options "SAMEORIGIN" always; add\_header X-XSS-Protection "1; mode=block" always; add\_header X-Content-Type-Options "nosniff" always; add\_header Referrer-Policy "no-referrer-when-downgrade" always; add\_header Content-Security-Policy "default-src 'self' http: https: data: blob: 'unsafe-inline'" always; # Logsaccess\_log /var/log/nginx/csdt-frontend.access.log; error\_log /var/log/nginx/csdt-frontend.error.log; }

### 4.2: Crear configuración para la API

# Crear archivo de configuraciónsudo nano /etc/nginx/sites-available/csdt-api

**Contenido del archivo:**

server { listen80; server\_name api-csdt.geekcorplab.com; # Proxy hacia el backend Laravellocation / { proxy\_pass http://127.0.0.1:8000; proxy\_http\_version1.1; 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\_timeout300s; proxy\_connect\_timeout75s; } # Configuración para archivos grandesclient\_max\_body\_size50M; # Logsaccess\_log /var/log/nginx/csdt-api.access.log; error\_log /var/log/nginx/csdt-api.error.log; }

### 4.3: Activar configuraciones de Nginx

# Crear enlaces simbólicossudoln -s /etc/nginx/sites-available/csdt-frontend /etc/nginx/sites-enabled/ sudoln -s /etc/nginx/sites-available/csdt-api /etc/nginx/sites-enabled/ # Eliminar sitio por defectosudorm /etc/nginx/sites-enabled/default # Verificar sintaxissudo nginx -t # Si está bien, recargar Nginxsudo systemctl reload nginx # Verificar estadosudo systemctl status nginx

## 🔒 PASO 5: CONFIGURAR FIREWALL

### 5.1: Configurar UFW

# Verificar estado actualsudo ufw status # Permitir SSHsudo ufw allow OpenSSH # Permitir HTTP y HTTPSsudo ufw allow 'Nginx Full'# Permitir puerto del backendsudo ufw allow 8000 # Activar firewallsudo ufw enable# Verificar estado finalsudo ufw status

## 🌐 PASO 6: CONFIGURAR DNS EN DIGITALOCEAN

### 6.1: En el panel de DigitalOcean

1. Ir a [DigitalOcean](https://cloud.digitalocean.com)
2. **Networking** → **Domains**
3. Agregar dominio: geekcorplab.com
4. Crear registros DNS:Tipo Nombre Valor A @ 64.225.113.49 A csdt 64.225.113.49 A api-csdt 64.225.113.49 CNAME www @

## 🔐 PASO 7: CONFIGURAR SSL CON LET'S ENCRYPT

### 7.1: Instalar Certbot

sudo apt install certbot python3-certbot-nginx -y

### 7.2: Obtener certificados SSL

# Para el frontendsudo certbot --nginx -d csdt.geekcorplab.com # Para la APIsudo certbot --nginx -d api-csdt.geekcorplab.com

## ✅ PASO 8: VERIFICACIÓN FINAL

### 8.1: Verificar servicios

# Verificar estado de Nginxsudo systemctl status nginx # Verificar estado de PM2sudo pm2 status # Verificar que la API responda curl http://localhost:8000 # Verificar logssudo pm2 logs csdt-backend

### 8.2: Verificar accesibilidad web

# Verificar que el frontend esté accesible curl -I http://csdt.geekcorplab.com # Verificar que la API esté accesible curl -I http://api-csdt.geekcorplab.com

### 8.3: Ver logs

# Logs de Nginxsudotail -f /var/log/nginx/csdt-frontend.access.log sudotail -f /var/log/nginx/csdt-api.access.log # Logs de PM2sudo pm2 logs csdt-backend # Logs de Laravelsudotail -f /var/www/csdt-backend/storage/logs/laravel.log

## 📁 ESTRUCTURA FINAL DEL SERVIDOR

/var/www/ ├── csdt-frontend/ # Frontend React/Vite │ ├── dist/ # Archivos compilados de Vite │ │ ├── index.html │ │ ├── assets/ │ │ └── ... │ ├── src/ # Código fuente │ │ ├── services/ # Servicios de IA (JavaScript) │ │ │ ├── IAMejoradaService.js │ │ │ ├── IAsProfesionalesService.js │ │ │ ├── SistemaIAProfesionalService.js │ │ │ ├── ChatGPTMejoradoService.js │ │ │ ├── IAsTecnicasService.js │ │ │ └── ConsejoIAService.js │ │ └── ... │ └── package.json └── csdt-backend/ # Backend Laravel ├── artisan # Comando principal ├── .env # Variables de entorno ├── composer.json # Dependencias PHP ├── package.json # Dependencias Node.js (Vite) ├── database/ │ ├── migrations/ # Migraciones de base de datos │ └── seeders/ # Seeders de datos ├── storage/ # Archivos de almacenamiento ├── vendor/ # Dependencias PHP └── node\_modules/ # Dependencias Node.js (Vite) /etc/nginx/ ├── sites-available/ │ ├── csdt-frontend # Configuración frontend │ └── csdt-api # Configuración API └── sites-enabled/ ├── csdt-frontend # Enlace simbólico └── csdt-api # Enlace simbólico

## 🔧 COMANDOS DE MANTENIMIENTO

### Actualizar frontend:

cd /var/www/csdt-frontend sudo git pull origin main sudo npm install sudo npm run build sudo systemctl reload nginx

### Actualizar backend:

cd /var/www/csdt-backend sudo git pull origin main sudo composer install --optimize-autoloader --no-dev sudo npm install # NO ejecutar npm run build aquí (solo en frontend)sudo php artisan migrate --force sudo php artisan config:cache sudo php artisan route:cache sudo php artisan view:cache sudo pm2 restart csdt-backend

### Ver estado de servicios:

# Estado de Nginxsudo systemctl status nginx # Estado de PM2sudo pm2 status # Estado del firewallsudo ufw status # Ver logssudo pm2 logs csdt-backend sudotail -f /var/www/csdt-backend/storage/logs/laravel.log

## 🎯 SERVICIOS DISPONIBLES

### URLs de Acceso:

* **Frontend:**https://csdt.geekcorplab.com
* **API Backend:**https://api-csdt.geekcorplab.com

### Puertos Internos:

* **Frontend:** 80/443 (Nginx)
* **Backend:** 8000 (Laravel/PM2)

## 🚨 SOLUCIÓN DE PROBLEMAS

### Si Nginx no inicia:

# Verificar sintaxissudo nginx -t # Ver logs de errorsudotail -f /var/log/nginx/error.log # Reiniciar Nginxsudo systemctl restart nginx

### Si PM2 no funciona:

# Ver estadosudo pm2 status # Ver logssudo pm2 logs csdt-backend # Reiniciarsudo pm2 restart csdt-backend

### Si Laravel no funciona:

# Verificar permisossudochown -R www-data:www-data /var/www/csdt-backend sudochmod -R 775 /var/www/csdt-backend/storage sudochmod -R 775 /var/www/csdt-backend/bootstrap/cache # Limpiar cachesudo php artisan config:clear sudo php artisan cache:clear sudo php artisan route:clear sudo php artisan view:clear

### Si MySQL no funciona:

# Verificar estado de MySQLsudo systemctl status mysql # Reiniciar MySQLsudo systemctl restart mysql # Verificar conexiónsudo mysql -u csdt -p -e "SHOW DATABASES;"# Verificar configuración de Laravelcd /var/www/csdt-backend sudo php artisan tinker # Dentro de tinker: DB::connection()->getPdo();# Salir: exit

### Si confundes XAMPP con MySQL:

**PROBLEMA:** "En mi computadora uso XAMPP, ¿por qué en DigitalOcean uso MySQL?"

**SOLUCIÓN:**

* **XAMPP:** Solo para desarrollo local en tu computadora
* **MySQL:** Para servidores de producción como DigitalOcean
* **Configuración local (.env):**DB\_CONNECTION=sqlite o DB\_CONNECTION=mysql (XAMPP)
* **Configuración producción (.env):**DB\_CONNECTION=mysql (servidor dedicado)

**NO mezcles:**

* ❌ No uses XAMPP en DigitalOcean
* ❌ No uses SQLite en producción
* ✅ Usa MySQL 8.0 en DigitalOcean
* ✅ Usa XAMPP solo en tu computadora local

### Si hay errores de build:

# Para el backend (NO ejecutar build):cd /var/www/csdt-backend sudo npm install # Para el frontend (SÍ ejecutar build):cd /var/www/csdt-frontend sudo npm install sudo npm run build

## 🔧 MEJORAS APLICADAS AL PROYECTO LOCAL

### Mejoras en IAMejoradaService.js:

* ✅ **Análisis especializado para Acción de Tutela** (Artículo 86 Constitución)
* ✅ **Análisis especializado para Acción Popular** (Artículo 88 Constitución)
* ✅ **Análisis especializado para Acción de Cumplimiento** (Artículo 87 Constitución)
* ✅ **Análisis especializado para Acción de Grupo**
* ✅ **Análisis especializado para Acción de Nulidad**
* ✅ **Análisis especializado para Acción de Reparación Directa**
* ✅ **Análisis especializado para Demanda Jurídica**
* ✅ **Recomendaciones específicas para cada tipo de acción**
* ✅ **Evaluación de viabilidad jurídica**
* ✅ **Estrategias procesales recomendadas**
* ✅ **Análisis de documentos y evidencias necesarias**
* ✅ **Cálculo de probabilidad de éxito**

### Integración con páginas existentes:

* ✅ **AccionTutela.jsx** - Integración completa con análisis de tutela
* ✅ **AccionPopular.jsx** - Integración completa con análisis popular
* ✅ **AccionCumplimiento.jsx** - Integración completa con análisis de cumplimiento
* ✅ **AccionGrupo.jsx** - Integración completa con análisis de grupo
* ✅ **AccionNulidad.jsx** - Integración completa con análisis de nulidad
* ✅ **AccionReparacionDirecta.jsx** - Integración completa con análisis de reparación
* ✅ **DemandaJuridica.jsx** - Integración completa con análisis de demandas

## 📝 NOTAS IMPORTANTES

1. **Servicios de IA:** Todos los servicios de IA están implementados en JavaScript y funcionan localmente en el frontend, incluyendo análisis especializados para las 7 acciones constitucionales implementadas
2. **Base de datos:** Configurado para usar MySQL 8.0 en producción (más robusto que SQLite)
3. **XAMPP vs MySQL:**
   * **XAMPP:** Solo para desarrollo local en tu computadora
   * **MySQL:** Para servidores de producción como DigitalOcean
   * **NO mezcles:** No uses XAMPP en DigitalOcean, no uses SQLite en producción
4. **Comandos de Build:**
   * Backend: NO ejecutar npm run build (solo instalar dependencias)
   * Frontend: SÍ ejecutar npm run build para compilar la aplicación
5. **Permisos:** Todos los archivos deben tener los permisos correctos para www-data
6. **Logs:** Monitorea regularmente los logs para detectar problemas
7. **Backup:** Configura backups regulares de la base de datos MySQL y archivos importantes
8. **Migraciones:** Siempre ejecutar php artisan migrate --force después de actualizaciones

## 🎯 RECOMENDACIONES ESPECÍFICAS PARA EL PROYECTO

### 1. **Integración de Servicios de IA con Páginas:**

// Ejemplo de integración en AccionTutela.jsximportIAMejoradaServicefrom'../services/IAMejoradaService'; // En el componenteconstanalizarConIA = async () => { const analisis = IAMejoradaService.analizarAccionTutela( formulario.hechos, formulario.derechoVulnerado, formulario.entidadDemandada ); setFormulario(prev => ({ ...prev, analisisIA: analisis })); };

### 2. **Configuración de Variables de Entorno:**

# Agregar al .env del frontend VITE\_IA\_ENABLED=true VITE\_IA\_ANALISIS\_ESPECIALIZADO=true VITE\_IA\_ACCIONES\_CONSTITUCIONALES=true

### 3. **Optimización de Performance:**

* Usar cache para análisis repetidos
* Implementar lazy loading para servicios de IA
* Optimizar imágenes y assets

### 4. **Seguridad:**

* Validar todas las entradas de usuario
* Implementar rate limiting para servicios de IA
* Usar HTTPS en producción

### 5. **Monitoreo:**

* Implementar logs detallados
* Monitorear uso de servicios de IA
* Configurar alertas de error

## 🎉 ¡INSTALACIÓN COMPLETADA!

Tu sistema CSDT está ahora completamente instalado y funcionando en DigitalOcean con:

* ✅ Frontend React/Vite con todos los servicios de IA en JavaScript
* ✅ Backend Laravel con Inertia.js y Vite
* ✅ 8 servicios de IA especializados implementados en JavaScript
* ✅ Análisis especializado para 7 acciones constitucionales (Tutela, Popular, Cumplimiento, Grupo, Nulidad, Reparación Directa, Demanda Jurídica)
* ✅ Base de datos MySQL 8.0 para producción
* ✅ Nginx como proxy reverso
* ✅ SSL con Let's Encrypt
* ✅ Firewall configurado
* ✅ PM2 para gestión de procesos
* ✅ Comandos de build corregidos (backend vs frontend)

¡El sistema está listo para usar en producción!