

INSTALACIÓN DE ALQUILOSCOOTER EN SERVIDOR PROPIO

ÍNDICE

-
1. [Requisitos del Servidor](#)
 2. [Instalación Paso a Paso](#)
 3. [Configuración de Base de Datos](#)
 4. [Variables de Entorno](#)
 5. [Instalación de Dependencias](#)
 6. [Compilación y Despliegue](#)
 7. [Configuración de Servicios Externos](#)
 8. [Troubleshooting](#)
-

REQUISITOS DEL SERVIDOR

Mínimos Recomendados:

- **Sistema Operativo:** Ubuntu 20.04+ o Debian 11+
- **CPU:** 4 cores
- **RAM:** 8 GB mínimo (16 GB recomendado)
- **Disco:** 50 GB SSD
- **Node.js:** v18.x o v20.x
- **PostgreSQL:** v14+ o v15+
- **Yarn:** v1.22+

Puertos Necesarios:

- **3000:** Aplicación Next.js (desarrollo)
- **80/443:** Nginx/Apache (producción)
- **5432:** PostgreSQL (interno)

INSTALACIÓN PASO A PASO

1. Preparar el Servidor

```
# Actualizar sistema  
sudo apt update && sudo apt upgrade -y  
  
# Instalar dependencias del sistema  
sudo apt install -y curl git build-essential
```

2. Instalar Node.js

```
# Instalar Node.js 20.x
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs

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

3. Instalar Yarn

```
# Instalar Yarn globalmente
npm install -g yarn

# Verificar instalación
yarn --version # Debe mostrar 1.22.x o superior
```

4. Instalar PostgreSQL

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

# Iniciar servicio
sudo systemctl start postgresql
sudo systemctl enable postgresql

# Verificar estado
sudo systemctl status postgresql
```

CONFIGURACIÓN DE BASE DE DATOS

1. Crear Base de Datos

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

# Dentro de psql, ejecutar:
CREATE DATABASE alquiloscooter_db;
CREATE USER alquiloscooter_user WITH PASSWORD 'TU_PASSWORD_SEGURA_AQUI';
GRANT ALL PRIVILEGES ON DATABASE alquiloscooter_db TO alquiloscooter_user;
\q
```

2. Configurar Acceso Remoto (si es necesario)

```
# Editar postgresql.conf
sudo nano /etc/postgresql/15/main/postgresql.conf

# Cambiar:
listen_addresses = 'localhost' # A:
listen_addresses = '*'

# Editar pg_hba.conf
sudo nano /etc/postgresql/15/main/pg_hba.conf

# Añadir:
host      all            all            0.0.0.0/0            md5

# Reiniciar PostgreSQL
sudo systemctl restart postgresql
```

VARIABLES DE ENTORNO

1. Subir Archivos al Servidor

```
# Subir el backup completo a tu servidor
scp -r alquiloscooter_backup_completo/* usuario@tu-servidor:/var/www/alquiloscooter/

# O usando FTP/SFTP con FileZilla, WinSCP, etc.
```

2. Configurar .env

```
cd /var/www/alquiloscooter

# Editar archivo .env
nano .env
```

3. Variables CRÍTICAS a Configurar:

```

# DATABASE (OBLIGATORIO)
DATABASE_URL="postgresql://alquiloscooter_user:TU_PASSWORD@localhost:5432/
alquiloscooter_db?schema=public"

# NEXTAUTH (OBLIGATORIO - Generar secreto único)
NEXTAUTH_SECRET="GENERA_UN_SECRET_ALEATORIO_AQUI"
NEXTAUTH_URL="https://tu-dominio.com"

# AWS S3 (OBLIGATORIO para fotos)
AWS_REGION="eu-west-3"
AWS_ACCESS_KEY_ID="tu_aws_access_key"
AWS_SECRET_ACCESS_KEY="tu_aws_secret_key"
AWS_BUCKET_NAME="tu-bucket-name"
AWS_FOLDER_PREFIX="alquiloscooter/"

# EMAIL SMTP (OBLIGATORIO)
SMTP_HOST="tu-servidor-smtp.com"
SMTP_PORT="587"
SMTP_USER="tu@email.com"
SMTP_PASSWORD="tu_password_email"
SMTP_FROM="Tu Empresa <tu@email.com>"
ADMIN_EMAIL="admin@tu-empresa.com"

# ABACUS AI (para OCR - OPCIONAL)
ABACUSAII_API_KEY="tu_api_key_abacus"

# GSCONTROL (Integración contable - OPCIONAL)
GSCONTROLL_API_KEY="tu_api_key_gscontrol"

# GOOGLE DRIVE (OPCIONAL)
GOOGLE_CLIENT_EMAIL="tu-service-account@proyecto.iam.gserviceaccount.com"
GOOGLE_PRIVATE_KEY="-----BEGIN PRIVATE KEY-----\nTU_PRIVATE_KEY\n-----END PRIVATE
KEY-----"
GOOGLE_DRIVE_FOLDER_ID="tu_folder_id"

```

4. Generar NEXTAUTH_SECRET:

```

# Generar secreto aleatorio
openssl rand -base64 32

# Copiar el resultado a NEXTAUTH_SECRET en .env

```

INSTALACIÓN DE DEPENDENCIAS

```

cd /var/www/alquiloscooter

# Instalar dependencias de Node
yarn install

# Esto puede tardar 5-10 minutos
# IMPORTANTE: Asegúrate de tener suficiente RAM (mínimo 4GB)

```

CONFIGURACIÓN DE PRISMA

1. Generar Cliente Prisma

```
cd /var/www/alquiloscooter

# Generar cliente Prisma
yarn prisma generate
```

2. Ejecutar Migraciones

```
# Aplicar migraciones a la base de datos
yarn prisma migrate deploy

# O si es la primera vez:
yarn prisma db push
```

3. Seed de Datos Iniciales (OPCIONAL)

```
# Poblar base de datos con datos iniciales
yarn prisma db seed
```

COMPILACIÓN Y DESPLIEGUE

1. Compilar Aplicación (Producción)

```
cd /var/www/alquiloscooter

# Compilar Next.js
NODE_OPTIONS="--max-old-space-size=6144" yarn build

# Esto puede tardar 5-15 minutos
```

2. Iniciar Aplicación

Opción A: Desarrollo (para pruebas)

```
yarn dev

# La app estará en http://localhost:3000
```

Opción B: Producción con PM2 (RECOMENDADO)

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

# Iniciar aplicación con PM2
pm2 start yarn --name "alquiloscooter" -- start

# Configurar inicio automático
pm2 startup
pm2 save

# Ver logs
pm2 logs alquiloscooter

# Reiniciar
pm2 restart alquiloscooter

# Detener
pm2 stop alquiloscooter
```

CONFIGURACIÓN DE NGINX (PRODUCCIÓN)

1. Instalar Nginx

```
sudo apt install -y nginx
```

2. Configurar Sitio

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

3. Contenido del Archivo:

```
server {
  listen 80;
  server_name tu-dominio.com www.tu-dominio.com;

  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_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;
  }
}
```

4. Activar Sitio

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

# Probar configuración
sudo nginx -t

# Reiniciar Nginx
sudo systemctl restart nginx
```

5. Instalar SSL con Let's Encrypt (RECOMENDADO)

```
# Instalar Certbot
sudo apt install -y certbot python3-certbot-nginx

# Obtener certificado SSL
sudo certbot --nginx -d tu-dominio.com -d www.tu-dominio.com

# Certbot configurará automáticamente HTTPS
# Los certificados se renovarán automáticamente
```



CONFIGURACIÓN DE SERVICIOS EXTERNOS

AWS S3 (Almacenamiento de Fotos)

1. **Crear cuenta AWS:** <https://aws.amazon.com/>
2. **Crear bucket S3:**
 - Región: eu-west-3 (París) o la más cercana
 - Permisos: Privado con acceso programático
3. **Crear usuario IAM** con permisos S3
4. **Copiar credenciales** a .env

SMTP (Email)

Opciones recomendadas:

- **Propio servidor:** Si tienes hosting con cPanel
- **Gmail:** Requiere "App Password"
- **SendGrid:** 100 emails/día gratis
- **Mailgun:** 5000 emails/mes gratis

Abacus AI (OCR - OPCIONAL)

1. Crear cuenta: <https://abacus.ai/>
2. Obtener API Key
3. Añadir a .env: ABACUSAI_API_KEY

Google Drive (Backup Automático - OPCIONAL)

1. Crear proyecto en Google Cloud Console
2. Activar Google Drive API
3. Crear Service Account
4. Descargar credenciales JSON

5. Extraer `client_email` y `private_key`

6. Añadir a `.env`

TROUBLESHOOTING

Error: “Cannot connect to database”

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

# Verificar credenciales en .env
# Probar conexión:
psql -h localhost -U alquiloscooter_user -d alquiloscooter_db
```

Error: “Out of memory”

```
# Aumentar límite de memoria para Node.js
NODE_OPTIONS="--max-old-space-size=8192" yarn build

# O añadir más RAM al servidor (mínimo 8GB)
```

Error: “Port 3000 already in use”

```
# Encontrar proceso usando el puerto
sudo lsof -i :3000

# Matar proceso
sudo kill -9 PID

# O cambiar puerto en package.json
```

Error: “Prisma Client not generated”

```
# Regenerar cliente Prisma
yarn prisma generate

# Si persiste, eliminar node_modules y reinstalar
rm -rf node_modules
yarn install
yarn prisma generate
```

Error: “Module not found”

```
# Reinstalar dependencias
rm -rf node_modules yarn.lock
yarn install
```



VERIFICACIÓN DE INSTALACIÓN

1. Comprobar que todo funciona:

```
# Base de datos
psql -h localhost -U alquiloscooter_user -d alquiloscooter_db -c "SELECT NOW();"

# Node.js
node --version

# Yarn
yarn --version

# PM2 (si se usa)
pm2 status

# Nginx (si se usa)
sudo systemctl status nginx
```

2. Acceder a la Aplicación:

- **Desarrollo:** <http://localhost:3000>
- **Producción:** <https://tu-dominio.com>

3. Login Inicial:

IMPORTANTE: Despues de la primera instalación, crear usuario admin:

```
cd /var/www/alquiloscooter

# Ejecutar script de creación de admin (si existe)
node scripts/create-admin.js

# O usar la interfaz de signup en /signup
```



MANTENIMIENTO

Backup de Base de Datos

```
# Crear backup
pg_dump -U alquiloscooter_user -h localhost alquiloscooter_db > backup_$(date + %Y%m%d).sql

# Restaurar backup
psql -U alquiloscooter_user -h localhost alquiloscooter_db < backup_20251125.sql
```

Actualizar Aplicación

```
cd /var/www/alquiloscooter

# Hacer backup antes de actualizar
pg_dump -U alquiloscooter_user alquiloscooter_db > backup_pre_update.sql

# Subir nuevos archivos
# ... (vía FTP/SCP)

# Instalar nuevas dependencias
yarn install

# Ejecutar migraciones
yarn prisma migrate deploy

# Recompilar
NODE_OPTIONS="--max-old-space-size=6144" yarn build

# Reiniciar aplicación
pm2 restart alquiloscooter
```

Logs

```
# Ver logs de PM2
pm2 logs alquiloscooter

# Ver logs de Nginx
sudo tail -f /var/log/nginx/access.log
sudo tail -f /var/log/nginx/error.log

# Ver logs de PostgreSQL
sudo tail -f /var/log/postgresql/postgres-15-main.log
```

SOPORTE

Si tienes problemas durante la instalación:

1. **Revisa los logs** de cada servicio
2. **Verifica las variables de entorno** en .env
3. **Comprueba los permisos** de archivos y carpetas
4. **Asegúrate de tener recursos suficientes** (RAM, CPU, disco)

CHECKLIST DE INSTALACIÓN

- [] Servidor con Ubuntu 20.04+ instalado
- [] Node.js 18.x o 20.x instalado
- [] Yarn instalado
- [] PostgreSQL instalado y configurado
- [] Base de datos creada

- [] Usuario de base de datos creado
 - [] Archivos del proyecto subidos al servidor
 - [] Archivo .env configurado con todas las variables
 - [] NEXTAUTH_SECRET generado
 - [] Dependencias instaladas (yarn install)
 - [] Cliente Prisma generado
 - [] Migraciones ejecutadas
 - [] Aplicación compilada (yarn build)
 - [] PM2 instalado y configurado
 - [] Aplicación iniciada con PM2
 - [] Nginx instalado (si producción)
 - [] Sitio Nginx configurado
 - [] SSL/HTTPS configurado con Certbot
 - [] AWS S3 configurado (credenciales en .env)
 - [] SMTP configurado (credenciales en .env)
 - [] Aplicación accesible desde navegador
 - [] Login funcionando
 - [] Backup de base de datos configurado
-

¡INSTALACIÓN COMPLETA! 🎉

Ahora tienes AlquiloScooter funcionando en tu propio servidor, completamente bajo tu control.