

# Guía de Deployment INMOVA a www.inmova.app

## Resumen Ejecutivo

Este documento proporciona una guía completa paso a paso para deployar la aplicación INMOVA a producción en [www.inmova.app](http://www.inmova.app).

**Estado actual:** La aplicación requiere ajustes de configuración antes del deployment.

## Pre-requisitos

Antes de comenzar el deployment, asegurar que:

1.  Tienes acceso a las claves de API de producción:
  - Stripe (claves live: `sk_live_*` y `pk_live_*`)
  - DocuSign (si aplica)
  - Redsys / Open Banking (si aplica)
  - SMTP / Email provider
2.  Tienes acceso a:
  - Base de datos de producción (PostgreSQL)
  - AWS S3 bucket para archivos
  - Dominio [www.inmova.app](http://www.inmova.app) configurado
3.  Has respaldado:
  - Base de datos actual
  - Archivos de configuración importantes

## Paso 1: Actualizar Variables de Entorno

### 1.1 Copiar template de producción

```
cp .env .env.backup  
cp .env.production.template .env.production
```

### 1.2 Editar .env con valores de producción

Abrir `.env` y actualizar las siguientes variables:

```
# IMPORTANTE: Actualizar con valores reales de producción

# 1. NextAuth
NEXTAUTH_URL=https://www.inmova.app
NEXTAUTH_SECRET=<GENERAR_NUEVO_CON: openssl rand -base64 32>

# 2. Stripe - CLAVES DE PRODUCCIÓN
STRIPE_SECRET_KEY=sk_live_<TU_CLAVE_SECRETA_PRODUCCION>
STRIPE_PUBLISHABLE_KEY=pk_live_<TU_CLAVE_PUBLICA_PRODUCCION>
STRIPE_WEBHOOK_SECRET=whsec_<TU_WEBHOOK_SECRET_PRODUCCION>
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=pk_live_<TU_CLAVE_PUBLICA_PRODUCCION>

# 3. Base de Datos de Producción
DATABASE_URL='postgresql://USER:PASSWORD@HOST:PORT/DATABASE?connect_timeout=15&pool_timeout=15&connection_limit=10'

# 4. AWS S3 - Verificar que apunta a bucket de producción
AWS_REGION=<TU_REGION>
AWS_BUCKET_NAME=<TU_BUCKET_PRODUCCION>
AWS_FOLDER_PREFIX=<TU_FOLDER_PREFIX>

# 5. Email / SMTP
SMTP_HOST=smtp.gmail.com
SMTP_PORT=587
SMTP_USER=<EMAIL_PRODUCCION>
SMTP_PASSWORD=<PASSWORD_0_APP_PASSWORD>
SMTP_FROM='INMOVA <noreply@inmova.app>'

# 6. Sentry (Monitoreo de Errores) - IMPORTANTE
SENTRY_DSN=<TU_SENTRY_DSN>
NEXT_PUBLIC_SENTRY_DSN=<TU_SENTRY_DSN>

# 7. Redis (Caching) - Opcional pero recomendado
UPSTASH_REDIS_REST_URL=<TU_UPSTASH_URL>
UPSTASH_REDIS_REST_TOKEN=<TU_UPSTASH_TOKEN>

# 8. Security Keys
CRON_SECRET=<GENERAR_CON: openssl rand -hex 32>
ENCRYPTION_KEY=<GENERAR_CON: openssl rand -hex 32>

# 9. Application
NODE_ENV=production
NEXT_PUBLIC_APP_URL=https://www.inmova.app
```

### 1.3 Generar claves seguras

```
# Generar NEXTAUTH_SECRET
openssl rand -base64 32

# Generar CRON_SECRET
openssl rand -hex 32

# Generar ENCRYPTION_KEY
openssl rand -hex 32
```

## Paso 2: Optimizar Configuración de Next.js

### 2.1 Reemplazar next.config.js con versión optimizada

```
# Backup del config actual
mv next.config.js next.config.js.backup

# Usar config optimizado
mv next.config.optimized.js next.config.js
```

Esto habilitará:

- Headers de seguridad HTTP
- Compresión de assets
- Optimización de imágenes
- Code splitting mejorado
- Remoción automática de console.log en producción

## Paso 3: Limpiar Console Statements (Opcional)

Si quieras limpiar manualmente los console statements antes del build:

```
# Ver qué se va a cambiar (dry run)
node scripts/clean-console-logs.js --dry-run

# Aplicar cambios
node scripts/clean-console-logs.js

# Verificar que no hay errores de tipos
yarn tsc --noEmit
```

**Nota:** El next.config.js optimizado ya remueve console statements automáticamente en el build de producción.

## Paso 4: Actualizar Base de Datos

### 4.1 Backup de base de datos actual

```
# Backup antes de migrar
pg_dump -h HOST -U USER -d DATABASE > backup_$(date +%Y%m%d_%H%M%S).sql
```

### 4.2 Generar cliente de Prisma

```
yarn prisma generate
```

## 4.3 Aplicar migraciones (si hay pendientes)

```
# Ver estado de migraciones
yarn prisma migrate status

# Aplicar migraciones pendientes
yarn prisma migrate deploy
```

## Paso 5: Ejecutar Tests y Verificaciones

### 5.1 Verificar preparación para producción

```
node scripts/check-production-readiness.js
```

Esto debe pasar sin errores críticos.

### 5.2 Verificar TypeScript

```
yarn tsc --noEmit
```

### 5.3 Ejecutar ESLint

```
yarn lint --fix
```

### 5.4 Ejecutar tests (si existen)

```
yarn test:ci
```

## Paso 6: Build de Producción

### 6.1 Limpiar builds anteriores

```
rm -rf .next
rm -rf .build
```

### 6.2 Ejecutar build

```
NODE_ENV=production yarn build
```

Esto debe completarse sin errores.

### 6.3 Verificar tamaño del bundle

```
node scripts/optimize-bundle.js
```

## 6.4 Analizar bundle (opcional)

```
# Ver distribución de archivos
du -h .next/static/chunks/* | sort -h | tail -20
```



## Paso 7: Deploy a www.inmova.app

### 7.1 Deploy usando herramienta de DeepAgent

```
# Desde el directorio del proyecto
cd /home/ubuntu/homming_vidaro

# Ejecutar deploy con hostname específico
# El deploy tool usará las variables de entorno de .env
```

Esto:

1. Empaquetará la aplicación
2. Creará bundle standalone
3. Subirá a servidores de producción
4. Configurará el dominio www.inmova.app

### 7.2 Configurar Stripe Webhooks

Después del deployment, configurar webhooks en Stripe:

1. Ir a: <https://dashboard.stripe.com/webhooks>
2. Crear nuevo endpoint: <https://www.inmova.app/api/webhooks/stripe>
3. Seleccionar eventos:
  - payment\_intent.succeeded
  - payment\_intent.payment\_failed
  - customer.subscription.created
  - customer.subscription.updated
  - customer.subscription.deleted
  - invoice.paid
  - invoice.payment\_failed
4. Copiar el **Signing Secret** y actualizar `STRIPE_WEBHOOK_SECRET` en `.env`



## Paso 8: Verificación Post-Deployment

### 8.1 Verificaciones Básicas

- La aplicación carga en <https://www.inmova.app>
- HTTPS está funcionando (certificado SSL válido)
- Redirects HTTP -> HTTPS funcionan

### 8.2 Verificaciones de Funcionalidad

- Login/Logout funciona correctamente
- Registro de nuevos usuarios funciona

- Reset de password funciona
- Dashboard carga correctamente
- Subida de archivos funciona (AWS S3)
- Emails se envían correctamente
- Pagos con Stripe funcionan (modo test primero)

### 8.3 Verificaciones de Performance

```
# Lighthouse audit
npx lighthouse https://www.inmova.app --view

# Web Vitals
# Verificar en DevTools -> Performance
```

Targets:

- FCP < 1.5s
- LCP < 2.5s
- TTI < 3.5s

### 8.4 Verificar Logs y Monitoreo

- Sentry está recibiendo eventos
- No hay errores críticos en logs
- Alertas configuradas correctamente

## Paso 9: Seguridad Post-Deployment

### 9.1 Configurar Rate Limiting

Verificar que rate limiting está activo en:

- /api/auth/\* (login, registro)
- /api/payments/\*
- APIs públicas

### 9.2 Configurar CORS (si aplica)

Verificar que solo dominios autorizados pueden acceder a las APIs.

### 9.3 Security Headers

Verificar headers en <https://securityheaders.com/?q=www.inmova.app>

Debe incluir:

- Strict-Transport-Security
- X-Frame-Options
- X-Content-Type-Options
- X-XSS-Protection
- Referrer-Policy

## 9.4 Audit de Seguridad

```
# Verificar vulnerabilidades
yarn audit --level high

# Actualizar dependencias con vulnerabilidades
yarn upgrade-interactive --latest
```



## Paso 10: Monitoreo y Mantenimiento

### 10.1 Configurar Alertas

- Errores críticos (Sentry)
- Downtime (UptimeRobot o similar)
- Uso de recursos (CPU, memoria, DB)
- Pagos fallidos (Stripe webhooks)

### 10.2 Backups Automáticos

Configurar backups diarios de:

- Base de datos (PostgreSQL)
- Archivos subidos (S3)
- Variables de entorno

### 10.3 Logs

Configurar rotación de logs:

- Retención: 30 días
- Compresión automática
- Archivado en S3



## Troubleshooting

### Problema: Build falla con errores de TypeScript

**Solución:**

```
# Ver errores detallados
yarn tsc --noEmit

# Limpiar cache
rm -rf .next node_modules/.cache
yarn install
```

### Problema: Imágenes no cargan

**Solución:**

- Verificar configuración de AWS S3
- Verificar que bucket tiene permisos públicos (solo para imágenes públicas)
- Verificar `next.config.js` tiene `remotePatterns` configurado

## Problema: Webhooks de Stripe no funcionan

### Solución:

- Verificar que `STRIPE_WEBHOOK_SECRET` es correcto
- Verificar que endpoint es accesible: <https://www.inmova.app/api/webhooks/stripe>
- Revisar logs de Stripe Dashboard

## Problema: Emails no se envían

### Solución:

- Verificar configuración SMTP
- Si usas Gmail, asegurar que tienes "App Password" configurado
- Verificar logs del servidor

## Problema: Performance lenta

### Solución:

- Verificar queries de base de datos (usar `EXPLAIN ANALYZE`)
- Activar Redis para caching
- Optimizar imágenes
- Lazy load de componentes pesados



## Recursos Adicionales

### Documentación

- Next.js Production: <https://nextjs.org/docs/deployment>
- Prisma Production: <https://www.prisma.io/docs/guides/deployment>
- Stripe Production: <https://stripe.com/docs/keys#test-live-modes>

### Herramientas de Monitoreo Recomendadas

- **Sentry** - Error tracking (ya instalado)
- **UptimeRobot** - Monitoreo de uptime
- **Datadog / New Relic** - APM y logs
- **LogRocket** - Session replay (opcional)

### Scripts Útiles

```
# Ver logs en tiempo real (si tienes acceso SSH)
tail -f /var/log/inmova/app.log

# Verificar estado del servidor
systemctl status inmova

# Restart aplicación
systemctl restart inmova

# Ver métricas de DB
psql -h HOST -U USER -d DATABASE -c "SELECT * FROM pg_stat_activity;"
```

## Checklist Final

---

Antes de considerar el deployment completo:

### Pre-Deployment

- [ ] Variables de entorno actualizadas con valores de producción
- [ ] Claves de Stripe son de PRODUCCIÓN (sk\_live\_, pk\_live\_)
- [ ] next.config.js optimizado implementado
- [ ] Database migrations aplicadas
- [ ] Backup de base de datos realizado
- [ ] `yarn build` completa sin errores
- [ ] `check-production-readiness.js` pasa sin errores críticos

### Durante Deployment

- [ ] Aplicación deployada a [www.inmova.app](http://www.inmova.app)
- [ ] SSL/HTTPS funcionando
- [ ] Redirects HTTP -> HTTPS configurados

### Post-Deployment

- [ ] Login/Logout funciona
- [ ] Registro de usuarios funciona
- [ ] Pagos con Stripe funcionan
- [ ] Emails se envían correctamente
- [ ] Subida de archivos funciona
- [ ] Webhooks de Stripe configurados
- [ ] Sentry recibiendo eventos
- [ ] Performance dentro de targets (Lighthouse)
- [ ] Security headers configurados
- [ ] Rate limiting activo
- [ ] Monitoreo y alertas configurados
- [ ] Backups automáticos configurados

---

## Contacto y Soporte

---

Para soporte durante el deployment:

- **Email Técnico:** [tech@inmova.app](mailto:tech@inmova.app)
  - **Documentación:** [docs.inmova.app](https://docs.inmova.app)
- 

**Última actualización:** 6 de Diciembre de 2025

**Versión:** 1.0

**Autor:** Equipo Técnico INMOVA