



Guía de Migración GitHub + Vercel - INMOVA



Estado Actual

✓ El proyecto está listo para migrar a GitHub y Vercel

Verificación Completada

- ✓ .gitignore configurado
 - ✓ .env.example creado
 - ✓ vercel.json configurado
 - ✓ package.json con scripts necesarios
 - ✓ Documentación completa
 - ✓ Prisma configurado
 - ✓ Estructura del proyecto correcta
 - ✓ Sin secretos hardcodeados
-



Archivos Importantes Creados

1. `.env.example` - Template de variables de entorno
 2. `vercel.json` - Configuración de Vercel
 3. `DEPLOYMENT.md` - Guía detallada de deployment
 4. `PRE-DEPLOYMENT-CHECKLIST.md` - Checklist de verificación
 5. `verify-deployment.sh` - Script de verificación automática
-

Pasos de Migración (Quick Start)

Paso 1: Preparar GitHub

```
# Asegúrate de estar en el directorio del proyecto
cd /home/ubuntu/homming_vidaro/nextjs_space

# Verificar estado de git
git status

# Agregar todos los cambios
git add .

# Commit
git commit -m "feat: ready for production deployment"

# Crear repositorio en GitHub (vía web: github.com/new)
# Luego conectar:
git remote add origin https://github.com/TU-USUARIO/inmova.git

# Push
git branch -M main
git push -u origin main
```

Paso 2: Configurar Servicios Externos

2.1. Base de Datos PostgreSQL

Recomendación: Vercel Postgres o Supabase

Opción A: Vercel Postgres

1. Dashboard de Vercel → Storage → Create Database
2. Seleccionar "Postgres"
3. Copiar `DATABASE_URL`

Opción B: Supabase

1. Crear proyecto en supabase.com (<https://supabase.com>)
2. Settings → Database → Connection String (Transaction Pooler)
3. Copiar la URL

Opción C: Neon

1. Crear cuenta en neon.tech (<https://neon.tech>)
2. Crear proyecto
3. Copiar Connection String

2.2. AWS S3 (Almacenamiento de Archivos)

1. Crear Bucket

- Ve a AWS S3 Console
- Create Bucket
- Nombre: `inmova-production` (o el que prefieras)
- Región: `us-west-2` (o la más cercana)

2. Configurar CORS


```
json
[
  {
```

```

    "AllowedHeaders": ["*"],
    "AllowedMethods": ["GET", "PUT", "POST", "DELETE"],
    "AllowedOrigins": ["https://tu-dominio.vercel.app", "https://inmova.app"],
    "ExposeHeaders": ["ETag"]
  }
]

```

3. Crear Usuario IAM

- IAM → Users → Create User
- Nombre: `inmova-s3-user`
- Attach Policy: `AmazonS3FullAccess` (o un policy personalizado más restrictivo)
- Crear Access Key
-  **Guardar:** `AWS_ACCESS_KEY_ID` y `AWS_SECRET_ACCESS_KEY`

2.3. Stripe (Pagos)

1. Crear cuenta en stripe.com (<https://stripe.com>)
2. Dashboard → Developers → API Keys
3. Obtener (modo Test primero):
 - `STRIPE_SECRET_KEY` (`sk_test_...`)
 - `STRIPE_PUBLISHABLE_KEY` (`pk_test_...`)
4. **Configurar Webhook** (después del primer deploy):
 - Developers → Webhooks → Add endpoint
 - URL: `https://tu-dominio.vercel.app/api/stripe/webhook`
 - Eventos: Seleccionar los necesarios
 - Copiar `STRIPE_WEBHOOK_SECRET`

2.4. NextAuth Secret

```

# Generar secret
openssl rand -base64 32

```

Paso 3: Deploy en Vercel

3.1. Conectar Repositorio

1. Ve a vercel.com (<https://vercel.com>) y haz login
2. “Add New” → “Project”
3. “Import Git Repository”
4. Selecciona tu repositorio de GitHub
5. Configuración:
 - **Framework Preset:** Next.js
 - **Root Directory:** `.` (raíz)
 - **Build Command:** `yarn build`
 - **Output Directory:** `.next`
 - **Install Command:** `yarn install`

3.2. Configurar Variables de Entorno

Antes de hacer deploy, agrega estas variables en Vercel:

Variables Requeridas (Production + Preview + Development)

```
# Base de Datos
DATABASE_URL=postgresql://usuario:password@host:5432/database

# NextAuth
NEXTAUTH_SECRET=<tu-secret-generado-con-openssl>
NEXTAUTH_URL=https://tu-dominio.vercel.app

# AWS S3
AWS_REGION=us-west-2
AWS_BUCKET_NAME=inmova-production
AWS_ACCESS_KEY_ID=AKIA...
AWS_SECRET_ACCESS_KEY=...
AWS_FOLDER_PREFIX=

# Stripe
STRIPE_SECRET_KEY=sk_test...
STRIPE_PUBLISHABLE_KEY=pk_test...
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=pk_test...

# Base URL
NEXT_PUBLIC_BASE_URL=https://tu-dominio.vercel.app

# Security
CRON_SECRET=<genera-uno-random>
ENCRYPTION_KEY=<genera-uno-random>

# Node
NODE_ENV=production
```

Variables Opcionales

```
# Stripe Webhook (agregar después de configurar webhook)
STRIPE_WEBHOOK_SECRET=whsec...

# Push Notifications (si aplica)
NEXT_PUBLIC_VAPID_PUBLIC_KEY=...
VAPID_PRIVATE_KEY=...

# Email (SendGrid)
SENDGRID_API_KEY=SG...
SENDGRID_FROM_EMAIL=noreply@inmova.app

# Monitoring (Sentry)
NEXT_PUBLIC_SENTRY_DSN=https://...@sentry.io/...

# Video
NEXT_PUBLIC_VIDEO_URL=https://www.youtube.com/embed/...
```

3.3. Deploy

1. Click en “Deploy”
2. Espera a que termine el build (2-5 minutos)
3. ¡Tu app estará live en `https://tu-proyecto.vercel.app` !

Paso 4: Post-Deployment

4.1. Ejecutar Migraciones de Prisma

Opción A: Desde tu máquina local

```
# 1. Descargar variables de entorno de Vercel
vercel env pull

# 2. Ejecutar migraciones
yarn prisma migrate deploy

# 3. (Opcional) Seed inicial
yarn prisma db seed
```

Opción B: Desde Vercel CLI

```
# Instalar Vercel CLI
npm i -g vercel

# Login
vercel login

# Link proyecto
vercel link

# Pull env vars
vercel env pull

# Run migrations
yarn prisma migrate deploy
```

4.2. Crear Usuario Administrador

Conecta a tu base de datos de producción y ejecuta:

```
-- 0 usa el script de seed
-- yarn prisma db seed
```

4.3. Configurar Webhook de Stripe

1. Stripe Dashboard → Developers → Webhooks
2. Add endpoint: `https://tu-dominio.vercel.app/api/stripe/webhook`
3. Seleccionar eventos:
 - `payment_intent.succeeded`
 - `payment_intent.failed`
 - `customer.subscription.created`
 - `customer.subscription.updated`
 - `customer.subscription.deleted`
4. Copiar Signing Secret
5. Agregarlo en Vercel como `STRIPE_WEBHOOK_SECRET`

4.4. Configurar Dominio Personalizado (Opcional)

1. Vercel Dashboard → Settings → Domains
2. Add Domain: `inmova.app`

3. Configurar DNS según instrucciones de Vercel
4. Esperar propagación DNS (puede tomar hasta 48h, generalmente <1h)
5. Actualizar `NEXTAUTH_URL` en Vercel: `https://inmova.app`
6. Actualizar CORS en AWS S3 con el nuevo dominio

4.5. Verificación Final

- ☐ La aplicación carga sin errores
- ☐ Login/Signup funcionan
- ☐ Dashboard se muestra correctamente
- ☐ Imágenes se cargan (S3)
- ☐ Base de datos conecta
- ☐ Creación de registros funciona
- ☐ Pagos funcionan (Stripe test mode)
- ☐ API endpoints responden
- ☐ No hay errores en console

Comandos Útiles

Local Development

```
# Instalar dependencias
yarn install

# Generar Prisma client
yarn prisma generate

# Ejecutar migraciones
yarn prisma migrate dev

# Seed database
yarn prisma db seed

# Desarrollo
yarn dev

# Build
yarn build

# Start production
yarn start
```

Vercel CLI

```
# Ver logs en tiempo real
vercel logs --follow

# Ver deployments
vercel list

# Ver variables de entorno
vercel env ls

# Agregar variable de entorno
vercel env add VARIABLE_NAME

# Pull env vars
vercel env pull
```

Prisma

```
# Ver base de datos en Prisma Studio
yarn prisma studio

# Crear migración
yarn prisma migrate dev --name nombre_migracion

# Aplicar migraciones en producción
yarn prisma migrate deploy

# Reset database (solo development)
yarn prisma migrate reset

# Generate Prisma Client
yarn prisma generate
```

Troubleshooting

Build Falla en Vercel

Error: Prisma Client no generado

```
// Verificar en package.json:
"scripts": {
  "postinstall": "prisma generate"
}
```

Error: Variables de entorno no definidas

- Verifica que todas las variables estén en Vercel
- Variables con `NEXT_PUBLIC` deben estar configuradas
- Redeploy después de agregar variables

Base de Datos no Conecta

1. Verificar que `DATABASE_URL` sea correcta
2. Verificar que el proveedor de DB permita conexiones desde Vercel

3. Para Supabase: usar Transaction Pooler URL
4. Para Neon: verificar que el endpoint esté activo

Imágenes no Cargan (S3)

1. Verificar CORS en S3
2. Verificar credenciales AWS
3. Verificar permisos del usuario IAM
4. Verificar que el bucket name sea correcto
5. Check browser console para errores específicos

Pagos no Funcionan (Stripe)

1. Verificar que las API keys sean del mismo ambiente (test o live)
2. Verificar webhook signature (STRIPE_WEBHOOK_SECRET)
3. Verificar eventos seleccionados en webhook
4. Ver logs en Stripe Dashboard → Developers → Webhooks

Error 500 en Producción

```
# Ver logs detallados
vercel logs tu-proyecto --follow

# O desde dashboard de Vercel
# Deployments → [tu deployment] → Logs
```



Monitoring y Mantenimiento

Analytics

- **Vercel Analytics:** Incluido automáticamente
- **Vercel Speed Insights:** Incluido automáticamente
- **Sentry:** Configurar `NEXT_PUBLIC_SENTRY_DSN`

Backups

Base de Datos

```
# Backup manual
pg_dump $DATABASE_URL > backup-$(date +%Y%m%d).sql

# Restore
psql $DATABASE_URL < backup-20231209.sql
```

Configurar backups automáticos en tu proveedor:

- Vercel Postgres: Backups automáticos
- Supabase: Settings → Database → Backups
- Neon: Backups automáticos incluidos

Actualizaciones

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

# Verificar vulnerabilidades
yarn audit

# Fix vulnerabilidades
yarn audit fix
```

Recursos

- [Vercel Documentation](https://vercel.com/docs) (https://vercel.com/docs)
- [Next.js Documentation](https://nextjs.org/docs) (https://nextjs.org/docs)
- [Prisma Documentation](https://www.prisma.io/docs) (https://www.prisma.io/docs)
- [Stripe Documentation](https://stripe.com/docs) (https://stripe.com/docs)
- [AWS S3 Documentation](https://docs.aws.amazon.com/s3/) (https://docs.aws.amazon.com/s3/)

Checklist Final

Antes del Deploy

- ☐ Código pusheado a GitHub
- ☐ .env.example actualizado
- ☐ Base de datos de producción creada
- ☐ AWS S3 bucket configurado
- ☐ Stripe account setup (modo test)
- ☐ Variables de entorno preparadas

Durante el Deploy

- ☐ Repositorio conectado en Vercel
- ☐ Variables de entorno configuradas
- ☐ Deploy exitoso
- ☐ Migraciones de Prisma ejecutadas
- ☐ Datos iniciales (seed) cargados

Después del Deploy

- ☐ Aplicación funciona correctamente
 - ☐ Usuario admin creado
 - ☐ Webhook de Stripe configurado
 - ☐ Dominio personalizado (si aplica)
 - ☐ Analytics configurado
 - ☐ Monitoring configurado
-

🎉 ¡Listo! Tu aplicación INMOVA estará corriendo en producción.

Para soporte adicional, consulta `DEPLOYMENT.md` o `PRE-DEPLOYMENT-CHECKLIST.md`.

Última actualización: Diciembre 2025

Estado: ☒ Verificado y listo para deployment