

Guía de Deployment en Vercel - INMOVA

Prerrequisitos

- Cuenta de Vercel (<https://vercel.com>)
- Cuenta de GitHub con el repositorio del proyecto
- Acceso a la base de datos PostgreSQL (Supabase o similar)
- Variables de entorno configuradas

Paso 1: Preparar el Repositorio en GitHub

1.1 Asegúrate de que el proyecto esté en GitHub

```
# Si aún no has inicializado git
cd /home/ubuntu/homming_vidaro
git init

# Agrega todos los archivos
git add .

# Haz el primer commit
git commit -m "Initial commit - INMOVA project"

# Conecta con tu repositorio de GitHub
git remote add origin https://github.com/TU_USUARIO/TU_REPO.git

# Sube los cambios
git push -u origin main
```

1.2 Archivos importantes ya configurados

- `vercel.json` - Configuración de Vercel
- `next.config.js` - Configuración de Next.js
- `.env` - Variables de entorno (NO subir a GitHub)
- `package.json` - Dependencias del proyecto

1.3 Crea un archivo `.gitignore` si no existe

```
# En el directorio raíz del proyecto
cat > .gitignore << 'EOF'
# dependencies
node_modules/
.pnp
.pnp.js
yarn-error.log

# testing
coverage/
.nyc_output

# next.js
.next/
out/
build/
dist/

# production
/build

# misc
.DS_Store
*.pem

# debug
npm-debug.log*
yarn-debug.log*
yarn-error.log*

# local env files
.env
.env.local
.env.development.local
.env.test.local
.env.production.local

# vercel
.vercel

# typescript
*.tsbuildinfo
next-env.d.ts

# prisma
prisma/migrations/

# IDE
.vscode/
.idea/
EOF
```

Paso 2: Configurar Vercel

2.1 Crear Nuevo Proyecto en Vercel

1. Ve a <https://vercel.com/new>
2. Selecciona “Import Git Repository”

3. Conecta tu cuenta de GitHub si no lo has hecho
4. Busca y selecciona tu repositorio `homming_vidaro`

2.2 Configuración del Proyecto

Framework Preset: Next.js

Root Directory: Deja en blanco o selecciona la carpeta raíz

Build Settings:

- Build Command: `cd nextjs_space && yarn build`
- Output Directory: `nextjs_space/.next`
- Install Command: `cd nextjs_space && yarn install`

2.3 Variables de Entorno

En la sección “Environment Variables” de Vercel, agrega todas estas variables:

Variables de Base de Datos

```
DATABASE_URL=postgresql://usuario:contraseña@host:5432/database
```

Variables de Autenticación

```
NEXTAUTH_SECRET=wJqizzZ073C6pU4tjLTNwzjeoGLaMWvr9
NEXTAUTH_URL=https://tu-dominio.vercel.app
```

Variables de AWS S3

```
AWS_PROFILE=hosted_storage
AWS_REGION=us-west-2
AWS_BUCKET_NAME=tu-bucket-name
AWS_FOLDER_PREFIX=tu-folder/
```

Variables de Stripe

```
STRIPE_SECRET_KEY=sk_test_tu_clave_aqui
STRIPE_PUBLISHABLE_KEY=pk_test_tu_clave_aqui
STRIPE_WEBHOOK_SECRET=whsec_tu_secret_aqui
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY=pk_test_tu_clave_aqui
```

Variables de Notificaciones Push

```
NEXT_PUBLIC_VAPID_PUBLIC_KEY=BEl62iUYgUivxIkv69yViEuiBIa-Ib27SzV9p3F-Jq-6-kxq9R-
wD9qdL4U3JfYxSh_Vu_WG2cEg8u7kJ7-vQTmE
VAPID_PRIVATE_KEY=p-K-PxeghWxVyGvxvHYVsT3xhp5fKWvUqNfNqN-J4XM
```

Variables de Abacus AI

```
ABACUSA1_API_KEY=a66d474df9e547058d3b977b3771d53b
```

Variables de Video

```
NEXT_PUBLIC_VIDEO_URL=https://www.youtube.com/embed/zm55Gdl5G1Q
```

Variables de Seguridad

```
CRON_SECRET=inmova-cron-secret-2024-secure-key-xyz789
ENCRYPTION_KEY=151b21e7b3a0ebb00a2ff5288f3575c9d4167305d3a84ccd385564955adefd2b
```

IMPORTANTE: Marca estas variables como disponibles para los 3 ambientes: Production, Preview y Development

■ **Paso 3: Configurar Base de Datos (Supabase)**

3.1 Crear Proyecto en Supabase

1. Ve a <https://supabase.com/dashboard>
2. Crea un nuevo proyecto
3. Anota la URL de conexión de PostgreSQL

3.2 Obtener la URL de Conexión

1. En tu proyecto de Supabase, ve a **Settings → Database**
2. Busca la sección “Connection string”
3. Selecciona “URI” y copia la cadena de conexión
4. Reemplaza `[YOUR-PASSWORD]` con tu contraseña

Ejemplo:

```
postgresql://postgres.xxxxx:TU_PASSWORD@aws-0-us-west-1.pooler.supabase.com:5432/postgres
```

3.3 Actualizar Variable de Entorno en Vercel

Actualiza `DATABASE_URL` en Vercel con la URL de Supabase.

3.4 Ejecutar Migraciones de Prisma

Después del primer deployment, necesitas ejecutar las migraciones:

```
# Opción 1: Desde tu máquina local
cd nextjs_space
DATABASE_URL="tu_url_de_supabase" npx prisma migrate deploy
DATABASE_URL="tu_url_de_supabase" npx prisma db seed

# Opción 2: Usar Vercel CLI
vercel env pull .env.local
yarn prisma migrate deploy
yarn prisma db seed
```

Paso 4: Integración con GitHub

4.1 Configurar Deploy Automático

Vercel automáticamente desplegará cuando:

- Hagas push a la rama `main` (producción)
- Hagas push a otras ramas (preview)
- Crees un Pull Request (preview)

4.2 Proteger Rama Principal

En GitHub, ve a **Settings → Branches → Add rule**:

- Branch name pattern: `main`
- Require pull request reviews before merging
- Require status checks to pass before merging
- Vercel – Production

4.3 Configurar Webhooks (Opcional)

Vercel configura automáticamente los webhooks necesarios con GitHub.

Paso 5: Deploy Inicial

5.1 Hacer Deploy

1. Haz clic en **Deploy** en Vercel
2. Espera a que se complete el build (puede tardar 5-10 minutos)
3. Vercel te dará una URL de producción: <https://tu-proyecto.vercel.app>

5.2 Verificar el Deploy

1. Ve a la URL proporcionada
2. Verifica que la aplicación cargue correctamente
3. Prueba el login con las credenciales:
 - **Super Admin:** `superadmin@inmova.com` / `superadmin123`
 - **Admin:** `admin@inmova.com` / `admin123`

Paso 6: Configuraciones Adicionales

6.1 Configurar Dominio Personalizado

1. En Vercel, ve a **Settings → Domains**
2. Agrega tu dominio: `inmova.app` o `www.inmova.app`
3. Sigue las instrucciones para configurar los registros DNS

6.2 Configurar Cron Jobs

Los cron jobs ya están configurados en `vercel.json`:

```

"crons": [
  {
    "path": "/api/cron/notifications",
    "schedule": "0 9 * * *"
  },
  {
    "path": "/api/cron/calendar-sync",
    "schedule": "0 */6 * * *"
  },
  {
    "path": "/api/cron/cleanup",
    "schedule": "0 2 * * 0"
  }
]

```

Nota: Los cron jobs solo funcionan en planes Pro de Vercel.

6.3 Configurar Analytics (Opcional)

1. Ve a **Analytics** en Vercel
2. Activa Vercel Analytics
3. Agrega el código en tu `app/layout.tsx` :

```

import { Analytics } from '@vercel/analytics/react';

export default function RootLayout({ children }) {
  return (
    <html>
      <body>
        {children}
        <Analytics />
      </body>
    </html>
  );
}

```

Troubleshooting

Error: “Module not found”

Solución:

```

# Verifica que todas las dependencias estén instaladas
cd nextjs_space
yarn install

```

Error: “Database connection failed”

Solución:

1. Verifica que `DATABASE_URL` esté correctamente configurada en Vercel
2. Asegúrate de que Supabase permita conexiones desde cualquier IP
3. En Supabase: **Settings → Database → Connection pooling → Enable**

Error: “Build failed”

Solución:

1. Revisa los logs en Vercel
2. Verifica que `next.config.js` esté correctamente configurado
3. Asegúrate de que `typescript.ignoreBuildErrors` esté en `false`

Error: “Prisma Client not generated”

Solución:

```
# Genera el cliente de Prisma localmente
cd nextjs_space
yarn prisma generate

# Agrega un build script en package.json
"postinstall": "prisma generate"
```



Monitoreo Post-Deploy

Verificar Métricas

1. **Build Time:** Debe ser < 5 minutos
2. **Function Execution:** Debe ser < 10 segundos
3. **Edge Network:** Debe usar la región más cercana

Logs y Debugging

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

# Login
vercel login

# Ver logs en tiempo real
vercel logs

# Ver logs de una función específica
vercel logs --since 1h
```



Workflow de Desarrollo

Desarrollo Local

```
cd nextjs_space
yarn dev
# Visita http://localhost:3000
```

Preview Deploy (Staging)

```
# Crea una rama de feature
git checkout -b feature/nueva-funcionalidad

# Haz cambios y commit
git add .
git commit -m "feat: nueva funcionalidad"

# Push a GitHub
git push origin feature/nueva-funcionalidad

# Vercel creará automáticamente un deploy de preview
```

Production Deploy

```
# Merge a main a través de Pull Request en GitHub
# O directamente:
git checkout main
git merge feature/nueva-funcionalidad
git push origin main

# Vercel desplegará automáticamente a producción
```

¡Deployment Completo!

Tu aplicación INMOVA ahora está desplegada en Vercel con:

- Deploy automático desde GitHub
- Base de datos PostgreSQL en Supabase
- Variables de entorno configuradas
- SSL/HTTPS automático
- CDN global
- Cron jobs configurados
- Dominios personalizados (opcional)

Soporte

Si necesitas ayuda adicional:

- **Vercel Documentation:** <https://vercel.com/docs>
- **Supabase Documentation:** <https://supabase.com/docs>
- **Next.js Documentation:** <https://nextjs.org/docs>

Última actualización: Diciembre 2024

Versión: 1.0

Proyecto: INMOVA - Sistema de Gestión Inmobiliaria