



Cambios Necesarios para Deployment en Vercel

Archivos a Modificar ANTES de hacer push a Git

1. Reemplazar `next.config.js`

Archivo actual: `nextjs_space/next.config.js`

Archivo nuevo: `nextjs_space/next.config.vercel.js`

```
cd /home/ubuntu/homming_vidaro/nextjs_space
cp next.config.js next.config.js.backup
cp next.config.vercel.js next.config.js
```

¿Por qué? El `next.config.js` actual tiene configuraciones específicas para el entorno de Abacus.Al que no son necesarias en Vercel.

2. Actualizar `package.json` - Build Command

NO es necesario modificar el `package.json`, pero en Vercel usará:

```
Build Command: yarn prisma generate && yarn build
```

Esto asegura que Prisma Client se genere antes del build.

3. Agregar `.vercelignore` (opcional pero recomendado)

Crear archivo `nextjs_space/.vercelignore`:

```
cat > /home/ubuntu/homming_vidaro/nextjs_space/.vercelignore << 'EOF'
node_modules
.next
.build
core
*.log
.env.local
.DS_Store
EOF
```

Archivos Nuevos Creados

- vercel.json** - Configuración de deployment
- .env.example** - Plantilla de variables de entorno
- next.config.vercel.js** - Next.js config optimizado para Vercel
- scripts/vercel-build.sh** - Script de build personalizado
- DEPLOYMENT_VERCEL.md** - Guía completa de deployment
- QUICK_START_VERCEL.md** - Guía rápida (15 minutos)
- VERCEL_MIGRATION_CHECKLIST.md** - Checklist paso a paso

Modificaciones Necesarias según tu Entorno

Si usas Base de Datos EXTERNA (recomendado)

No hay cambios necesarios. Simplemente asegúrate de que:

1. La base de datos sea accesible desde internet
2. El firewall permita conexiones desde las IPs de Vercel
3. `DATABASE_URL` sea correcta

Si usas Vercel Postgres

1. En Vercel, ve a **Storage → Create Database → Postgres**
2. Vercel te dará una `DATABASE_URL` automáticamente
3. No necesitas configurarla manualmente

Si NO usas AWS S3 (usas otro storage)

Modifica en `lib/aws-config.ts` y `lib/s3.ts` para usar tu proveedor:

- Cloudflare R2
- Google Cloud Storage
- Azure Blob Storage
- Vercel Blob

O usa **Vercel Blob** (recomendado):

```
yarn add @vercel/blob
```

Y modifica las funciones de upload/download.

Comandos Pre-Deployment

```
cd /home/ubuntu/homming_vidaro/nextjs_space

# 1. Hacer backup del next.config.js actual
cp next.config.js next.config.js.backup

# 2. Usar la versión para Vercel
cp next.config.vercel.js next.config.js

# 3. Crear .vercelignore
cat > .vercelignore << 'EOF'
node_modules
.next
.build
core
*.log
.env.local
.DS_Store
EOF

# 4. Verificar que todo compile
yarn prisma generate
yarn build

# 5. Si el build es exitoso, commit y push
git add .
git commit -m "Configuración para deployment en Vercel"
git push origin main
```

Diferencias entre Abacus.AI y Vercel

Aspecto	Abacus.AI	Vercel
Output	Standalone	Automatic
Build Dir	.build	.next
Env Vars	.env file	Dashboard UI
Database	Hosted DB	External/Vercel Postgres
Storage	AWS S3	AWS S3 / Vercel Blob
Deploy	CLI Tool	Git Push
Domains	subdomain.abacusai.app	custom domains gratis
SSL	Auto	Auto
CDN	Included	Included (Edge Network)
Logs	File based	Real-time Dashboard

Verificación Pre-Deploy

Checklist Rápido:

```
# 1. Verificar que Prisma funciona
yarn prisma generate

# 2. Verificar que el build funciona
yarn build

# 3. Verificar variables de entorno críticas
grep -E '(DATABASE_URL|NEXTAUTH_SECRET|AWS_BUCKET_NAME)' .env

# 4. Verificar que el .gitignore está correcto
cat .gitignore

# 5. Verificar que no hay archivos grandes (> 100MB)
find . -type f -size +100M
```

Post-Deployment

1. Primera Verificación (Inmediata)

```
# Probar la URL de Vercel
curl -I https://tu-proyecto.vercel.app

# Debería devolver HTTP 200
```

2. Verificación de Features

- [] Login/Logout
- [] Crear edificio
- [] Subir imagen
- [] Ver dashboard
- [] Crear contrato
- [] Procesar pago (si aplica)

3. Monitoreo (Primeras 24 horas)

1. **Runtime Logs:** Ve a Vercel → Deployments → Runtime Logs
2. **Analytics:** Ve a Vercel → Analytics
3. **Errores:** Busca errores 500 o 404

Troubleshooting

Build Fails con “Cannot find module”

```
# Solución: Verificar package.json
# Asegúrate de que todas las dependencias estén en dependencies, no devDependencies
```

Build Fails con “Prisma generate failed”

```
# Solución: Modificar vercel.json
{
  "buildCommand": "yarn prisma generate --no-hints && yarn build"
}
```

Runtime Error: “Database connection failed”

```
# Solución: Verificar que DATABASE_URL sea accesible
# Prueba desde tu terminal:
psql "postgresql://
role_587683780:5kWw7vKJBDp9ZA2Jfkt5BdWrAjR0XDe5@db-587683780.db003.hosteddb.reai.io:
5432/587683780"

# Si no conecta, verifica firewall/permisos
```

Imágenes no se ven

```
# Solución 1: Verificar AWS credentials
# En Vercel, añade:
AWS_ACCESS_KEY_ID=tu_access_key
AWS_SECRET_ACCESS_KEY=tu_secret_key

# Solución 2: Verificar permisos del bucket S3
# El bucket debe tener política que permita acceso público para archivos públicos
```

“NEXTAUTH_URL is not defined”

```
# Solución: Añadir variable en Vercel
# Settings → Environment Variables
NEXTAUTH_URL=https://tu-proyecto.vercel.app
# o
NEXTAUTH_URL=https://inmova.app
```

Rollback

Si algo sale mal y necesitas volver a la versión anterior:

1. En Vercel, ve a **Deployments**
2. Encuentra el deployment anterior que funcionaba
3. Click en los 3 puntos → **Promote to Production**

O en Git:

```
git revert HEAD
git push origin main
```

Migración Completa vs Prueba

Opción 1: Migración Completa (Producir en inmova.app)

1. Seguir QUICK_START_VERCEL.md
2. Configurar dominio custom
3. Actualizar DNS
4. Migrar usuarios

Opción 2: Deploy de Prueba (Primero en Vercel URL)

1. Deploy en Vercel sin dominio custom
2. Probar en <https://tu-proyecto.vercel.app>
3. Verificar todo funciona
4. Luego configurar dominio custom

Recomendación: Empezar con Opción 2 para evitar downtime.

Sopporte

- **Documentación Vercel:** <https://vercel.com/docs>
- **Vercel Support:** support@vercel.com
- **Status:** <https://www.vercel-status.com/>
- **Community:** <https://github.com/vercel/vercel/discussions>

Última actualización: 5 de diciembre de 2024