

Guía Rápida: Implementación de Optimizaciones



Implementación en 5 Minutos



Estado Actual

✓ Archivos preparados:

- `next.config.optimized.js` - Configuración optimizada de Next.js
 - `vercel.json` - Configuración de despliegue
 - `OPTIMIZACIONES_BUILD.md` - Documentación completa
-

Opción 1: Aplicación Automática (Recomendado)

Paso 1: Ejecutar el Script de Aplicación

```
cd /home/ubuntu/homming_vidaro
chmod +x aplicar_optimizaciones.sh
./aplicar_optimizaciones.sh
```

El script:

1. ✓ Hace backup del `next.config.js` actual
2. ✓ Aplica la configuración optimizada
3. ✓ Instala dependencias si es necesario
4. ✓ Verifica la configuración

Paso 2: Probar Build

```
cd nextjs_space
yarn build
```

Paso 3: Analizar Resultados

```
ANALYZE=true yarn build
```

Esto abrirá un reporte visual en tu navegador.

Opción 2: Aplicación Manual

Paso 1: Backup

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

Paso 2: Aplicar Configuración

```
cp ../next.config.optimized.js next.config.js
```

Paso 3: Verificar

```
node -e "console.log(require('../next.config.js'))"
```

Paso 4: Build de Prueba

```
yarn build
```



Verificación de Mejoras

Antes vs Después

Después del build, compara:

Antes:

Route (app)	Size	First Load
└ ○ /	5.2 kB	850 kB ✗ Grande
└ ○ /dashboard	12 kB	880 kB ✗ Grande

Después (Esperado):

Route (app)	Size	First Load
└ ○ /	5.2 kB	450 kB ✓ Optimizado
└ ○ /dashboard	12 kB	480 kB ✓ Optimizado

+ Chunks:		
└ framework.js	45 kB	✓
└ ui-lib.js	38 kB	✓
└ chart-lib.js	42 kB	✓
└ commons.js	50 kB	✓

Métricas Clave

Métrica	Objetivo	Cómo Verificar
Chunks > 244KB	< 3 chunks	Revisar output del build
First Load	< 500KB	Ver "First Load JS"
Build Time	< 5 min	Tiempo total de build
Lighthouse	> 85	lighthouse https://inmova.app

Troubleshooting Rápido

Problema: Build falla

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

# Reinstalar dependencias
rm -rf node_modules .next
yarn install
yarn build
```

Problema: Chunks aún grandes

```
# Analizar bundle
ANALYZE=true yarn build

# Buscar bibliotecas problemáticas
# En el reporte, identificar módulos > 244KB
```

Problema: Error de timeout en despliegue

1. Verificar `vercel.json` existe en la raíz
2. En Vercel Dashboard:
 - Settings → General
 - Build & Development Settings
 - Build Command Timeout: 300 segundos

Checklist Post-Implementación

- ☐ Build local exitoso
 - ☐ Chunks < 244KB
 - ☐ First Load < 500KB
 - ☐ Analizar con `ANALYZE=true yarn build`
 - ☐ Probar en desarrollo: `yarn dev`
 - ☐ Desplegar a staging
 - ☐ Verificar Lighthouse score
 - ☐ Monitorear errores en producción
-

Comandos Útiles

```
# Build y analizar
ANALYZE=true yarn build

# Solo analizar browser
ANALYZE=true BUNDLE_ANALYZE=browser yarn build

# Solo analizar server
ANALYZE=true BUNDLE_ANALYZE=server yarn build

# Build con más memoria
NODE_OPTIONS="--max-old-space-size=4096" yarn build

# Limpiar cache y rebuild
rm -rf .next && yarn build
```



Próximos Pasos

Corto Plazo (Hoy)

1. Aplicar optimizaciones
2. Verificar build local
3. Analizar bundle

Medio Plazo (Esta Semana)

1. Desplegar a staging
2. Monitorear métricas
3. A/B testing si es necesario

Largo Plazo (Este Mes)

1. Configurar monitoreo continuo
2. Establecer presupuesto de performance
3. Auditorías mensuales



Documentación Adicional

- **Detalles completos:** Ver `OPTIMIZACIONES_BUILD.md`
- **Configuración optimizada:** Ver `next.config.optimized.js`
- **Configuración Vercel:** Ver `vercel.json`



Preguntas Frecuentes

Q: ¿Puedo revertir los cambios?

A: Sí, usa el backup: `cp next.config.js.backup next.config.js`

Q: ¿Cómo sé si funcionó?

A: Compara el tamaño de “First Load JS” antes y después

Q: ¿Qué hago si el build falla?

A: Restaura el backup y reporta el error con el log completo

Q: ¿Necesito cambiar código de la aplicación?

A: No, solo cambios en configuración. Pero para mejor tree shaking, usa named imports.

Última actualización: Diciembre 2024

Versión: 1.0.0