



Comandos de Mantenimiento INMOVA

🧹 Limpieza de Cachés

Limpieza Básica (Segura)

```
# Limpiar cachés de build
find .next -type f -delete 2>/dev/null || true
find .build -type f -delete 2>/dev/null || true
find .swc -type f -delete 2>/dev/null || true

# Limpiar caché de Yarn
yarn cache clean

# Limpiar archivos de log
find . -name "*.log" -type f -delete 2>/dev/null || true
```

Limpieza Completa con Script

```
# Ejecutar script de limpieza automatizado
bash scripts/clean-caches.sh
```



Optimización de Base de Datos

Análisis de Base de Datos

```
# Ejecutar análisis completo
yarn tsx --require dotenv/config scripts/optimize-database.ts
```

Mantenimiento de Prisma

```
# Validar esquema
yarn prisma validate

# Ver estado de migraciones
yarn prisma migrate status

# Aplicar migraciones pendientes
yarn prisma migrate deploy

# Regenerar cliente de Prisma
yarn prisma generate
```

Optimización de PostgreSQL

```
# VACUUM y ANALYZE (mejora rendimiento de queries)
psql $DATABASE_URL -c "VACUUM ANALYZE;"

# Ver estadísticas de tablas
psql $DATABASE_URL -c "SELECT schemaname, tablename, n_live_tup, n_dead_tup FROM pg_stat_user_tables ORDER BY n_live_tup DESC;"

# Ver tamaño de tablas
psql $DATABASE_URL -c "SELECT table_name,
pg_size.pretty(pg_total_relation_size(quote_ident(table_name))) AS size FROM information_schema.tables WHERE table_schema = 'public' ORDER BY pg_total_relation_size(quote_ident(table_name)) DESC;"
```



Gestión de Dependencias

Análisis de Dependencias

```
# Detectar dependencias no utilizadas
npx depcheck --ignores="@types/*,eslint-*,prettier,@testing-library/*,@storybook/*,@playwright/*,vitest,jest"

# Listar todas las dependencias
yarn list --depth=0

# Ver dependencias duplicadas
yarn dedupe --check
```

Eliminar Dependencias No Usadas

⚠ PRECAUCIÓN: Prueba en una rama separada antes de aplicar en producción.

```
# Respaldar package.json
cp package.json package.json.backup

# Eliminar dependencias probablemente seguras
yarn remove @dnd-kit/core @dnd-kit/sortable @dnd-kit/utilities \
  formik gray-matter i18next i18next-browser-languagedetector \
  react-i18next isomorphic-dompurify mapbox-gl next-intl \
  swagger-jsdoc tailwind-scrollbar-hide yup

# Eliminar devDependencies no usadas
yarn remove -D ignore-loader null-loader ts-jest

# Reinstalar y probar
yarn install
yarn test:ci
```



Flujo de Mantenimiento Completo

Mantenimiento Semanal

```

#!/bin/bash
set -e

echo "✍️ Iniciando mantenimiento semanal..."

# 1. Limpiar cachés
echo "1. Limpmando cachés..."
bash scripts/clean-caches.sh

# 2. Analizar base de datos
echo "2. Analizando base de datos..."
yarn tsx --require dotenv/config scripts/optimize-database.ts

# 3. Validar Prisma
echo "3. Validando Prisma..."
yarn prisma validate

# 4. Ejecutar tests
echo "4. Ejecutando tests..."
yarn test:ci

echo "✅ Mantenimiento completado!"

```

Mantenimiento Mensual

```

#!/bin/bash
set -e

echo "✍️ Iniciando mantenimiento mensual..."

# 1. Mantenimiento semanal
bash scripts/weekly-maintenance.sh

# 2. Analizar dependencias
echo "2. Analizando dependencias..."
npx depcheck --ignores="@types/*,eslint-*,prettier" > dependency-report.txt

# 3. Optimizar PostgreSQL
echo "3. Optimizando PostgreSQL..."
psql $DATABASE_URL -c "VACUUM ANALYZE;"

# 4. Backup de base de datos (si está configurado)
echo "4. Creando backup..."
# pg_dump $DATABASE_URL > backup-$(date +%Y%m%d).sql

echo "✅ Mantenimiento mensual completado!"

```



Monitoreo de Rendimiento

Métricas de Build

```
# Tiempo de build
time yarn build

# Tamaño de bundles
du -sh .next/static/*

# Análisis de bundle (requiere next-bundle-analyzer)
yarn build && yarn analyze
```

Métricas de Base de Datos

```
# Queries lentas (en PostgreSQL)
psql $DATABASE_URL -c "SELECT query, calls, total_time, mean_time FROM pg_stat_statements ORDER BY mean_time DESC LIMIT 10;"

# Conexiones activas
psql $DATABASE_URL -c "SELECT count(*) FROM pg_stat_activity;"

# Tamaño de base de datos
psql $DATABASE_URL -c "SELECT pg_size_pretty(pg_database_size(current_database()));"
```



Troubleshooting

Problemas Comunes

Build falla después de limpieza

```
# Regenerar todo
yarn prisma generate
rm -rf .next .build
yarn build
```

Error de Prisma Client

```
# Regenerar cliente
yarn prisma generate

# Si persiste, reinstalar
rm -rf node_modules
yarn install
yarn prisma generate
```

Base de datos lenta

```
# Optimizar PostgreSQL
psql $DATABASE_URL -c "VACUUM FULL ANALYZE;"
psql $DATABASE_URL -c "REINDEX DATABASE nombre_bd;"
```

Espacio en disco lleno

```
# Ver uso de disco  
du -sh .next .build .swc node_modules  
  
# Limpiar todo  
bash scripts/clean-caches.sh  
yarn cache clean  
  
# Verificar logs del sistema  
du -sh /var/log/*
```



Recursos

- [Next.js - Building Your Application](https://nextjs.org/docs/building-your-application) (<https://nextjs.org/docs/building-your-application>)
- [Prisma - Optimization](https://www.prisma.io/docs/guides/performance-and-optimization) (<https://www.prisma.io/docs/guides/performance-and-optimization>)
- [PostgreSQL - Routine Database Maintenance Tasks](https://www.postgresql.org/docs/current/maintenance.html) (<https://www.postgresql.org/docs/current/maintenance.html>)
- [Yarn - CLI Commands](https://classic.yarnpkg.com/en/docs/cli/) (<https://classic.yarnpkg.com/en/docs/cli/>)

Última actualización: 5 de Diciembre 2025