

Reporte de Limpieza y Optimización INMOVA

Fecha: 5 de Diciembre 2025

Proyecto: /home/ubuntu/homming_vidaro



Resumen Ejecutivo

Espacio en Disco

- **.next (Build cache):** 61 MB
- **.build (Standalone build):** 2.8 GB 
- **.swc (Compiler cache):** 12 KB
- **Total cachés:** ~2.86 GB

Dependencias

- **Total de paquetes instalados:** 2,007
- **Dependencias no utilizadas detectadas:** 35
- **DevDependencies no utilizadas:** 12



Acciones Realizadas

1. Limpieza de Caché de Yarn

- Cache de Yarn limpiado exitosamente

2. Análisis de Dependencias

Se detectaron las siguientes dependencias que aparentemente no están en uso:

Dependencies no utilizadas (35):

```
@dnd-kit/core
@dnd-kit/sortable
@dnd-kit/utilities
@floating-ui/react
@hookform/resolvers
@tanstack/react-virtual
autoprefixer
cookie
csv
dayjs
formik
gray-matter
i18next
i18next-browser-languagedetector
isomorphic-dompurify
jotai
lodash
mapbox-gl
next-intl
plotly.js
qs
rate-limiter-flexible
react-datepicker
react-i18next
react-intersection-observer
react-plotly.js
react-select
swagger-jsdoc
swr
tailwind-scrollbar-hide
winston
yup
zustand
```

DevDependencies no utilizadas (12):

```
@next/swc-wasm-nodejs
@tanstack/react-query-devtools
@typescript-eslint/eslint-plugin
@typescript-eslint/parser
eslint
ignore-loader
jest-environment-jsdom
null-loader
postcss
ts-jest
ts-node
tsx
```

Recomendaciones Importantes

Dependencias a Revisar Manualmente

Algunas dependencias marcadas como “no utilizadas” pueden estar en uso de forma indirecta:

1. **autoprefixer** - Usado por PostCSS/Tailwind
2. **@hookform/resolvers** - Validación de formularios con React Hook Form
3. **lodash** - Utilidades comunes (verificar imports)
4. **dayjs** - Manipulación de fechas (si no se usa date-fns en todo el proyecto)
5. **zustand** - Gestión de estado (verificar si realmente no se usa)
6. **plotly.js / react-plotly.js** - Gráficos (verificar páginas de analytics/BI)

Dependencias Probablemente Seguras para Eliminar

```
{
  "probablementeSeguros": [
    "@dnd-kit/*",
    "formik",
    "gray-matter",
    "i18next",
    "i18next-browser-languagedetector",
    "react-i18next",
    "isomorphic-dompurify",
    "mapbox-gl",
    "next-intl",
    "swagger-jsdoc",
    "tailwind-scrollbar-hide",
    "yup"
  ]
}
```

Optimización de Base de Datos

Índices Recomendados

Se recomienda crear índices en las siguientes tablas para mejorar el rendimiento:

```

// User
@@index([email])
@@index([companyId])
@@index([role])

// Building
@@index([companyId])
@@index([direccion])

// Unit
@@index([buildingId])
@@index([estado])

// Contract
@@index([unitId])
@@index([tenantId])
@@index([fechaInicio])
@@index([fechaFin])

// Payment
@@index([contractId])
@@index([estado])
@@index([fechaVencimiento])

// MaintenanceRequest
@@index([buildingId])
@@index([unitId])
@@index([estado])
@@index([prioridad])

```



Comandos de Limpieza Recomendados

1. Limpiar Build Caches (Seguro)

```

cd /home/ubuntu/homming_vidaro/nextjs_space

# Limpiar Next.js cache
rm -rf .next/*

# Limpiar build standalone
rm -rf .build/*

# Regenerar Prisma Client
yarn prisma generate

# Rebuild limpio
yarn build

```

2. Optimizar Base de Datos

```
# Verificar esquema
yarn prisma validate

# Ver cambios pendientes
yarn prisma migrate status

# Aplicar migraciones
yarn prisma migrate deploy

# Opcional: Resetear y sembrar (SÓLO EN DEV)
# yarn prisma migrate reset
```

3. Eliminar Dependencias No Usadas (PRECAUCIÓN)

⚠️ IMPORTANTE: Prueba en una rama separada primero.

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

# Eliminar dependencias 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

# Ejecutar tests
yarn test:ci

# Si todo funciona, eliminar devDependencies no usadas
yarn remove -D ignore-loader null-loader ts-jest

# Reinstalar
yarn install
```



Impacto Esperado

Espacio en Disco

- **Reducción de cachés:** ~2.8 GB liberados (después de limpieza de .build)
- **Reducción de node_modules:** ~50-100 MB (eliminando dependencias no usadas)
- **Total liberado:** ~2.9 GB

Rendimiento

- **Build time:** Mejora del 10-15% (menos dependencias)
- **Queries DB:** Mejora del 30-50% (con índices adecuados)
- **Bundle size:** Reducción del 5-10% (eliminando imports no usados)

Mantenimiento Continuo

Scripts Recomendados para package.json

```
{
  "scripts": {
    "clean": "rm -rf .next .build .swc",
    "clean:all": "yarn clean && rm -rf node_modules yarn.lock",
    "clean:build": "rm -rf .next .build && yarn prisma generate",
    "analyze": "yarn depcheck --ignores='@types/*,eslint-* ,prettier-'",
    "db:optimize": "yarn prisma migrate deploy && yarn prisma generate",
    "optimize": "yarn clean && yarn install && yarn db:optimize && yarn build"
  }
}
```

Checklist Mensual

- [] Limpiar cachés de build (`.next` , `.build`)
- [] Ejecutar `yarn analyze` para detectar dependencias no usadas
- [] Revisar logs de performance de queries DB
- [] Ejecutar `yarn prisma migrate status`
- [] Verificar espacio en disco del servidor

Soporte

Para dudas sobre esta limpieza:

- **Email:** `tech@inmova.com`
- **Documentación:** `/docs/maintenance.md`

Generado automáticamente el 5 de Diciembre 2025