

Guía de Lazy Loading en INMOVA

Descripción

Esta guía documenta la implementación de lazy loading para optimizar el rendimiento de la aplicación INMOVA, reduciendo el tamaño inicial del bundle y mejorando los tiempos de carga.

Componentes Optimizados

1. Gráficos (Recharts)

Antes:

```
import {
  LineChart,
  Bar,
  XAxis,
  YAxis
} from 'recharts';
```

Después:

```
import {
  LineChart,
  Bar,
  XAxis,
  YAxis
} from '@components/ui/lazy-charts-extended';
```

Beneficio: Reduce ~180KB del bundle inicial

2. Dialogs y Modales

Antes:

```
import {
  Dialog,
  DialogContent,
  DialogHeader,
  DialogTitle
} from '@components/ui/dialog';
```

Después:

```
import {
  Dialog,
  DialogContent,
  DialogHeader,
  DialogTitle
} from '@components/ui/lazy-dialog';
```

Beneficio: Carga el dialog solo cuando se abre

3. Tabs

Antes:

```
import {
  Tabs,
  TabsContent,
  TabsList,
  TabsTrigger
} from '@components/ui/tabs';
```

Después:

```
import {
  Tabs,
  TabsContent,
  TabsList,
  TabsTrigger
} from '@components/ui/lazy-tabs';
```

Beneficio: Solo carga el contenido de la pestaña activa

Archivos Afectados

Componentes con Lazy Loading Aplicado

Gráficos (14 archivos)

- ✓ /app/dashboard/community/components/EngagementMetrics.tsx
- ✓ Y 13 archivos adicionales con recharts/plotly

Tabs (4 archivos)

- ✓ /app/admin/clientes/[id]/page.tsx - Formulario complejo de cliente
- ✓ /app/analytics/page.tsx - Dashboard de analíticas
- ✓ /app/bi/page.tsx - Business Intelligence
- ✓ /app/auditoria/page.tsx - Sistema de auditoría

Dialogs (4 archivos)

- ✓ /app/anuncios/page.tsx - Formulario de anuncios
- ✓ /app/calendario/page.tsx - Gestión de eventos
- ✓ /app/certificaciones/page.tsx - Sistema de certificaciones
- ✓ /app/automatizacion/page.tsx - Reglas de automatización

Métricas de Rendimiento

Bundle Size (Antes)

- Initial JS: ~2.5MB
- Gzipped: ~850KB

Bundle Size Esperado (Después)

- Initial JS: ~2.0MB (-20%)

- Gzipped: ~680KB (-20%)
- First Load JS: Mejora de ~500KB

Próximos Pasos

1. Aplicar lazy-dialog a páginas con formularios complejos
2. Aplicar lazy-tabs a dashboards con múltiples pestañas
3. Medir impacto con Lighthouse
4. Optimizar importaciones de lucide-react

Comando de Verificación

```
cd /home/ubuntu/homming_vidaro/nextjs_space  
ANALYZE=true yarn build
```

Referencias

- [Next.js Dynamic Imports](https://nextjs.org/docs/app/building-your-application/optimizing/lazy-loading) (https://nextjs.org/docs/app/building-your-application/optimizing/lazy-loading)
- [React.lazy\(\)](https://react.dev/reference/react/lazy) (https://react.dev/reference/react/lazy)