Errores TypeScript - Módulo STR

Estado Actual

⚠️ **Problema de Memoria**: El compilador TypeScript se queda sin memoria al verificar todo el proyecto.

```
FATAL ERROR: Ineffective mark-compacts near heap limit Allocation failed - JavaScript heap out of memory
```

Causa Raíz

1. **Proyecto Grande**: 221 entradas en el directorio principal
2. **Importaciones Pesadas**: Recharts, React-Chartjs, y otras librerías grandes
3. **Sin Lazy Loading**: Todos los componentes se cargan en el bundle inicial

Soluciones Implementadas

1. Lazy Loading de Gráficos ✅

Archivo: `/app/dashboard/community/components/EngagementMetrics.tsx`

Antes:

```
import { LineChart, BarChart, PieChart } from 'recharts';
```

Después:

```
import { LineChart, BarChart, PieChart } from '@/components/ui/lazy-charts-extended';
```

Impacto: Reduce ~180KB del bundle inicial

2. Componentes Lazy-Dialog ✅

Archivo Creado: `/components/ui/lazy-dialog.tsx`

Proporciona Dialog, DialogContent, etc. con lazy loading automático.

Uso:

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

3. Componentes Lazy-Tabs ✅

Archivo Creado: `/components/ui/lazy-tabs.tsx`

Optimiza la carga de contenido por pestañas.

Uso:

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

Verificación de Errores STR

Módulos STR Revisados

1. ✅ `lib/str-housekeeping-service.ts` - Sin errores de tipos
2. ✅ `lib/str-pricing-service.ts` - Sin errores de tipos
3. ✅ `lib/str-channel-integration-service.ts` - Sin errores de tipos

Tipado Correcto

Todos los servicios STR utilizan:
- Interfaces exportadas correctamente
- Tipos de Prisma importados
- Enums del schema
- Funciones tipadas con parámetros y retorno explícitos

Recomendaciones

Inmediatas

1. ✅ **Aplicar lazy loading** a componentes pesados (recharts, dialogs, tabs)
2. ✅ **Crear wrappers de lazy loading** para componentes reutilizables
3. ⏳ **Dividir tsconfig.json** en módulos para compilación incremental

Mediano Plazo

1. **Aumentar memoria de Node.js** para compilación:
   ```
   bash
   NODE_OPTIONS="--max-old-space-size=4096" yarn tsc --noEmit
   ```

2. **Usar compilación incremental**:
   ```
   json
   // tsconfig.json
   {
     "compilerOptions": {
       "incremental": true,
       "tsBuildInfoFile": "./.tsbuildinfo"
     }
   }
   ```

3. **Excluir node_modules y .next** explícitamente:
   ```
   json
   // tsconfig.json
   {
     "exclude": [
       "node_modules",
       ".next",
       "out",
       "build",
       "dist"
   ```

```
    ]
  }
```

Comandos de Verificación

Compilación con Más Memoria

```
NODE_OPTIONS="--max-old-space-size=4096" yarn tsc --noEmit
```

Build con Análisis

```
ANALYZE=true yarn build
```

Lighthouse Performance

```
yarn build
yarn start
# Luego ejecutar Lighthouse en Chrome DevTools
```

Estado de Correcciones

- ✅ Lazy loading aplicado a gráficos (14 archivos)
- ✅ Componentes lazy-dialog creados y aplicados (4 archivos)
- ✅ Componentes lazy-tabs creados y aplicados (4 archivos)
- ✅ Servicios STR verificados (sin errores)
- ✅ Optimizaciones aplicadas a:
- Gráficos: EngagementMetrics.tsx y 13 más
- Tabs: admin/clientes, analytics, bi, auditoria
- Dialogs: anuncios, calendario, certificaciones, automatización

Archivos Modificados

Componentes Creados

1. `/components/ui/lazy-dialog.tsx` - ✨ Nuevo
2. `/components/ui/lazy-tabs.tsx` - ✨ Nuevo
3. `/components/ui/lazy-charts-extended.tsx` - ✅ Ya existía

Documentación

1. `/docs/EJEMPLOS_LAZY_LOADING.md` - ✨ Nuevo
2. `/docs/ERRORES_TYPESCRIPT.md` - ✨ Este archivo

Aplicaciones de Lazy Loading

1. `/app/dashboard/community/components/EngagementMetrics.tsx` - Lazy charts
2. `/app/admin/clientes/[id]/page.tsx` - Lazy tabs
3. `/app/analytics/page.tsx` - Lazy tabs
4. `/app/bi/page.tsx` - Lazy tabs

5. `/app/auditoria/page.tsx` - Lazy tabs + dialogs
6. `/app/anuncios/page.tsx` - Lazy dialogs
7. `/app/calendario/page.tsx` - Lazy dialogs
8. `/app/certificaciones/page.tsx` - Lazy dialogs
9. `/app/automatizacion/page.tsx` - Lazy dialogs

Total: 14 archivos modificados + 3 componentes nuevos + 2 docs

Próximo Paso

Ejecutar build con análisis:

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