

Implementación de SEO Técnico - Fase 1

Resumen

Se ha implementado un sistema completo de SEO técnico para INMOVA con las siguientes características:

✓ Funcionalidades Implementadas

1. Meta-tags Dinámicos (Open Graph + Twitter Cards)

Ubicación: `lib/seo-utils.ts`

- **Función** `generateMetaTags()` : Genera meta-tags completos para cualquier página
- **Función** `generatePropertyMetaTags()` : Genera meta-tags específicos para propiedades
- **Soporte completo de Open Graph**: Título, descripción, imágenes, tipo de contenido, locale
- **Twitter Cards**: Configuración de `summary_large_image` con imágenes optimizadas
- **URLs Canónicas**: Para evitar contenido duplicado

Ejemplo de uso:

```
import { generatePropertyMetaTags } from '@lib/seo-utils';

export async function generateMetadata({ params }) {
  const property = await fetchProperty(params.id);
  return generatePropertyMetaTags({
    id: property.id,
    titulo: property.titulo,
    descripcion: property.descripcion,
    precio: property.precio,
    // ... más datos
  });
}
```

2. Structured Data (JSON-LD)

Ubicación: `lib/structured-data.ts`

- **Organization Schema**: Datos de la organización INMOVA
- **Property Schema**: Datos estructurados para propiedades (Product/RealEstateListing)
- **Breadcrumb Schema**: Navegación jerárquica
- **Componente StructuredDataScript**: Para insertar JSON-LD en el head

Schemas implementados:

- `generateOrganizationSchema()` : Info de INMOVA
- `generatePropertySchema()` : Detalles de propiedad con precio, ubicación, características
- `generateBreadcrumbSchema()` : Rutas de navegación

Ejemplo de uso:

```
import { StructuredDataScript } from '@components/seo/StructuredDataScript';
import { generatePropertySchema } from '@lib/structured-data';

const schema = generatePropertySchema(propertyData);

return (
  <>
    <StructuredDataScript data={schema} />
    { /* contenido */ }
  </>
);
```

3. Sitemap Dinámico

Ubicación: app/sitemap.ts

- **Generación automática:** Se actualiza dinámicamente con las propiedades de la DB
- **URLs estáticas:** Páginas principales del sitio
- **URLs dinámicas:**
 - Propiedades/unidades (hasta 1000)
 - Edificios (hasta 500)
- **Prioridades optimizadas:** Homepage (1.0), propiedades (0.7), etc.
- **Frecuencia de cambio:** Weekly, monthly según tipo de contenido
- **Filtrado inteligente:** Solo propiedades disponibles y edificios activos

Acceso: <https://inmova.app/sitemap.xml>

4. Robots.txt

Ubicación: app/robots.ts

- **Acceso permitido:** Páginas públicas y propiedades
- **Acceso bloqueado:**
 - APIs internas
 - Dashboard y admin
 - Páginas de edición
 - Archivos Next.js internos
- **Bots bloqueados:** GPTBot, ChatGPT, Claude, etc. (protección de contenido)
- **Referencia al sitemap:** Link automático

Acceso: <https://inmova.app/robots.txt>

5. Botones de Compartir en Redes Sociales

Ubicación: components/ui/share-buttons.tsx

Componentes:

- **ShareButtons** : Botones completos para compartir
- Modos: `inline` (botones visibles) o `dropdown` (menú desplegable)
- Redes: Facebook, Twitter/X, LinkedIn, WhatsApp, Email
- Botón de copiar enlace al portapapeles
- Tamaños: sm, md, lg
- **SimpleShareButton** : Botón simple con Web Share API
- Usa API nativa del navegador si está disponible

- Fallback a copiar al portapapeles

Ejemplo de uso:

```
import { ShareButtons } from '@components/ui/share-buttons';

<ShareButtons
  url={propertyUrl}
  title="Propiedad en Madrid"
  description="Hermoso apartamento de 2 habitaciones"
  hashtags={['inmova', 'alquiler', 'madrid']}
  variant="inline"
  size="sm"
/>
```

6. Preview de Redes Sociales

Ubicación: components/seo/SocialPreview.tsx

Componentes:

- **SocialPreview** : Preview completo con tabs
 - Visualización de Facebook, Twitter, LinkedIn
 - Simulación realista de cómo se verá el contenido
 - Botón de mostrar/ocultar
 - Info de optimización SEO
- **SocialPreviewCompact** : Versión compacta
 - Thumbnail + texto
 - Badges de redes sociales

Ejemplo de uso:

```
import { SocialPreview } from '@components/seo/SocialPreview';

<SocialPreview
  title="Mi Propiedad"
  description="Descripción de la propiedad"
  url="https://inmova.app/unidades/123"
  image="https://i.ytimg.com/vi/Ngm7-2sqw3s/mqdefault.jpg"
  siteName="INMOVA"
/>
```

📁 Estructura de Archivos

app/		
[-] sitemap.ts	# Sitemap dinámico	
[-] robots.ts	# Robots.txt dinámico	
[-] layout.tsx	# Layout principal con meta-tags globales	
[-] unidades/[id]/		
[-] layout.tsx	# Layout con SEO específico de unidad	
[-] page.tsx	# Página con ShareButtons y SocialPreview	
lib/		
[-] seo-utils.ts	# Utilidades para meta-tags	
[-] structured-data.ts	# Generadores de JSON-LD schemas	
[-] seo-config.ts	# Configuración global de SEO	
components/		
[-] ui/		
[-] share-buttons.tsx	# Botones de compartir	
[-] seo/		
[-] SocialPreview.tsx	# Preview de redes sociales	
[-] StructuredDataScript.tsx	# Componente para JSON-LD	
public/		
[-] inmovi-og-image.jpg	# Imagen Open Graph por defecto (1200x630)	
[-] inmovi-property-default.jpg	# Imagen por defecto para propiedades	

⚙️ Configuración

Variables de entorno requeridas:

```
NEXT_PUBLIC_BASE_URL=https://inmovi.app
```

Imágenes requeridas:

1. /public/inmovi-og-image.jpg - 1200x630px (Open Graph)
2. /public/inmovi-property-default.jpg - Imagen por defecto para propiedades sin foto
3. /public/inmovi-logo.png - Logo de INMOVA

🚀 Integración en Páginas

Página de Detalle de Unidad

Ya implementado en `app/unidades/[id]/page.tsx` :

1. **Meta-tags dinámicos**: Generados en `layout.tsx`
2. **Structured Data**: Inyectado automáticamente
3. **Botones de compartir**: En sección principal
4. **Preview social**: Debajo de info principal

Cómo Añadir SEO a Otras Páginas

1. Para páginas de servidor (Server Components):

```
// app/mi-pagina/page.tsx
import { Metadata } from 'next';
import { generateMetaTags } from '@lib/seo-utils';

export const metadata: Metadata = generateMetaTags({
  title: 'Mi Página',
  description: 'Descripción de mi página',
  url: 'https://inmova.app/mi-pagina',
  images: ['/mi-imagen.jpg'],
  type: 'website',
});

export default function MiPagina() {
  return <div>Contenido</div>;
}
```

2. Para páginas dinámicas:

```
// app/mi-pagina/[id]/layout.tsx
import { Metadata } from 'next';

interface Props {
  params: { id: string };
  children: React.ReactNode;
}

export async function generateMetadata({ params }: Props): Promise<Metadata> {
  const data = await fetchData(params.id);
  return generateMetaTags({
    title: data.title,
    description: data.description,
    url: `https://inmova.app/mi-pagina/${params.id}`,
    images: data.images,
  });
}

export default function Layout({ children }: Props) {
  return <>{children}</>;
}
```

3. Añadir Structured Data:

```
import { StructuredDataScript } from '@components/seo/StructuredDataScript';
import { generateOrganizationSchema } from '@lib/structured-data';

export default function MiPagina() {
  const schema = generateOrganizationSchema();

  return (
    <>
      <StructuredDataScript data={schema} />
      {/* contenido */}
    </>
  );
}
```

Mejores Prácticas

Meta-tags

- **Título:** 50-60 caracteres
- **Descripción:** 150-160 caracteres
- **Imágenes OG:** 1200x630px (ratio 1.91:1)
- **Imágenes Twitter:** 1200x628px o similar

Structured Data

- Siempre validar con [Google Rich Results Test](https://search.google.com/test/rich-results) (<https://search.google.com/test/rich-results>)
- Usar tipos específicos según contenido (Product, RealEstateListing, Organization, etc.)
- Incluir todos los campos recomendados por schema.org

Sitemap

- Actualizar frecuencias realistas (no daily si no cambia diariamente)
- Prioridades lógicas (homepage = 1.0, páginas secundarias < 1.0)
- Limitar a URLs importantes (evitar millones de URLs)

Testing y Validación

Herramientas recomendadas:

1. Meta-tags:

- [Meta Tags](https://metatags.io/) (<https://metatags.io/>)
- [OpenGraph Preview](https://www.opengraph.xyz/) (<https://www.opengraph.xyz/>)

2. Structured Data:

- [Google Rich Results Test](https://search.google.com/test/rich-results) (<https://search.google.com/test/rich-results>)
- [Schema Markup Validator](https://validator.schema.org/) (<https://validator.schema.org/>)

3. Sitemap:

- [XML Sitemaps Validator](https://www.xml-sitemaps.com/validate-xml-sitemap.html) (<https://www.xml-sitemaps.com/validate-xml-sitemap.html>)

4. Robots.txt:

- [Google Robots Testing Tool](https://support.google.com/webmasters/answer/6062598) (<https://support.google.com/webmasters/answer/6062598>)

5. Social Sharing:

- [Facebook Sharing Debugger](https://developers.facebook.com/tools/debug/) (<https://developers.facebook.com/tools/debug/>)
- [Twitter Card Validator](https://cards-dev.twitter.com/validator) (<https://cards-dev.twitter.com/validator>)
- [LinkedIn Post Inspector](https://www.linkedin.com/post-inspector/) (<https://www.linkedin.com/post-inspector/>)

Comandos Útiles

Ver sitemap generado:

```
curl https://inmova.app/sitemap.xml
```

Ver robots.txt:

```
curl https://inmova.app/robots.txt
```

Validar meta-tags en navegador:

```
// Abrir consola del navegador y ejecutar:
document.querySelectorAll('meta[property^="og:"]');
document.querySelectorAll('meta[name^="twitter:"]');
```

Troubleshooting

Problema: Imágenes no aparecen en previews sociales

- Verificar que las imágenes sean accesibles públicamente
- Usar URLs absolutas (no relativas)
- Verificar tamaño mínimo (1200x630px recomendado)
- Usar [Facebook Debugger](https://developers.facebook.com/tools/debug/) (https://developers.facebook.com/tools/debug/) para limpiar caché

Problema: Sitemap no se genera

- Verificar que Prisma pueda conectarse a la DB
- Revisar logs de servidor para errores
- Verificar permisos de lectura en tablas

Problema: Structured Data no valida

- Usar [Google Rich Results Test](https://search.google.com/test/rich-results) (https://search.google.com/test/rich-results)
- Verificar que todos los campos requeridos estén presentes
- Revisar tipos de datos (string, number, etc.)

Métricas de Éxito

KPIs a monitorear:

- Posicionamiento en Google (ranking de keywords)
- CTR (Click-Through Rate) en resultados de búsqueda
- Shares en redes sociales
- Tráfico orgánico desde Google
- Apariciones en rich snippets

Herramientas:

- Google Search Console
- Google Analytics
- Social media analytics

Próximas Mejoras (Fases Futuras)

- **Fase 2:** Análisis de keywords y contenido optimizado
- **Fase 3:** Link building interno y externo
- **Fase 4:** Performance y Core Web Vitals
- **Fase 5:** Contenido multimedia optimizado
- **Fase 6:** Internacionalización y hreflang

Última actualización: Diciembre 2024

Versión: 1.0

Autor: DeepAgent - Abacus.AI