

⚡ PLAN DE ACCIÓN INMEDIATO - GRUASEQUISER.COM

SIGUIENTE PASO: OPTIMIZACIÓN DE IMÁGENES

Fecha: 15 de Diciembre, 2025



ESTADO ACTUAL



COMPLETADO

- Routing de blogs corregido (/blog/[slug])
- Sitemap.xml completo (107 URLs)
- Robots.txt configurado
- Metadatos SEO completos (Open Graph, Twitter Card)
- Schema.org JSON-LD implementado
- Verificación noindex (solo en 404/error)
- Build exitoso (176 páginas estáticas)
- Deploy en producción (gruasequier.com)



PENDIENTE (PRIORIDAD ALTA)

- Optimizar TOP 10 imágenes críticas (30MB → 5MB)
- Decidir sobre sistema de routing dual
- Enviar sitemap a Google Search Console
- Solicitar indexación prioritaria



OBJETIVO INMEDIATO

Meta: Aumentar PageSpeed Mobile de **77/100** a **90+**

Estrategia: Optimizar TOP 10 imágenes más pesadas

Ahorro esperado: 30MB → 5MB = **83% de reducción**

Tiempo estimado: **2 horas**

Impacto: **+10-13 puntos en PageSpeed**



PASO 1: OPTIMIZAR TOP 10 IMÁGENES CRÍTICAS

Imágenes a Optimizar (Prioridad MÁXIMA)

#	Archivo Origin-al	Tamaño	Objetivo	Acción
1	trabajo grua 800 ton.png	8.5MB	1.2MB	WebP 85%
2	movilizacion-topas-metro-ca-racas.png	8.5MB	1.2MB	WebP 85%
3	movilizacion-generador-sobredimension-ado.png	3.2MB	450KB	WebP 85%
4	logo-equiser-actualizado.png	3.0MB	400KB	WebP 90%
5	logo equiser actulizado sin fondo.png	3.0MB	350KB	WebP 90%
6	trabajo estadio copa amer-ica.png	2.6MB	400KB	WebP 85%
7	trabajo gruas de 600 ton de-mag.png	2.5MB	380KB	WebP 85%
8	dos gruas de 600 ton.png	2.3MB	350KB	WebP 85%
9	trabajo de grua.png	2.2MB	330KB	WebP 85%
10	movilizacion-vagones-ferro-carril.jpg	2.2MB	330KB	WebP 85%

MÉTODO A: Optimización Manual con Squoosh.app (RECOMENDADO)

Ventajas:

- Control total sobre calidad visual
- No requiere instalación
- Interfaz visual comparativa
- Gratis

Proceso Paso a Paso:

Para CADA una de las 10 imágenes:

1. Abrir Squoosh.app

```
https://squoosh.app
```

2. Arrastrar imagen PNG al navegador

```
Imagen: /home/ubuntu/gruas_equierer_website/app/public/images/trabajo_grua_800_ton.png
```

3. Configurar panel derecho (Compress):

```
Format: WebP
Quality: 85% (imágenes de proyectos)
          90% (logos - #4 y #5)
Effort: 6 (máxima compresión)
Resize: NO (mantener tamaño original)
```

4. Comparar visualmente:

- Squoosh muestra split view (antes/después)
- Verificar que no haya pérdida significativa de calidad
- Ajustar quality si es necesario (80-90%)

5. Descargar imagen optimizada:

- Clic en botón **Download** (esquina inferior derecha)
- Guardar con nombre normalizado:

```
trabajo_grua_800_ton.png → trabajo-grua-800-ton.webp
```

6. Subir a servidor:

```
# Desde tu máquina local (donde descargaste la imagen)
scp ~/Downloads/trabajo-grua-800-ton.webp ubuntu@server:/home/ubuntu/gruas_equierer_website/app/public/images/
```

```
# O copiar manualmente si tienes acceso al servidor
```

7. Repetir para las 10 imágenes

Tiempo estimado por imagen: 10 minutos

Tiempo total: 100 minutos (1h 40min)

MÉTODO B: Script de Conversión Masiva (MÁS RÁPIDO)

Ventajas:

- Automático para todas las imágenes
- Más rápido (30 min vs 100 min)
- Procesa múltiples imágenes en paralelo

Desventajas:

- Menos control visual
- Requiere instalación de sharp-cli

Comandos:

```

# 1. Instalar sharp-cli
cd /home/ubuntu/gruas_equier_website/app
yarn add -D sharp-cli

# 2. Crear directorio para imágenes optimizadas
mkdir -p public/images/optimized

# 3. Convertir TOP 10 imágenes
cd public/images

# Imagen 1: trabajo grua 800 ton.png
npx sharp-cli "trabajo grua 800 ton.png" \
-o "optimized/trabajo-grua-800-ton.webp" \
--quality 85 --progressive

# Imagen 2: movilizacion-topas-metro-caracas.png
npx sharp-cli "movilizacion-topas-metro-caracas.png" \
-o "optimized/movilizacion-topas-metro-caracas.webp" \
--quality 85 --progressive

# Imagen 3: movilizacion-generador-sobredimensionado.png
npx sharp-cli "movilizacion-generador-sobredimensionado.png" \
-o "optimized/movilizacion-generador-sobredimensionado.webp" \
--quality 85 --progressive

# Imagen 4: logo-equier-actualizado.png
npx sharp-cli "logo-equier-actualizado.png" \
-o "optimized/logo-equier-actualizado.webp" \
--quality 90 --progressive

# Imagen 5: logo equiser actulizado sin fondo.png
npx sharp-cli "logo equiser actulizado sin fondo.png" \
-o "optimized/logo-equier-actualizado-sin-fondo.webp" \
--quality 90 --progressive

# Imagen 6: trabajo estadio copa america.png
npx sharp-cli "trabajo estadio copa america.png" \
-o "optimized/trabajo-estadio-copa-america.webp" \
--quality 85 --progressive

# Imagen 7: trabajo gruas de 600 ton demag.png
npx sharp-cli "trabajo gruas de 600 ton demag.png" \
-o "optimized/trabajo-gruas-600-ton-demag.webp" \
--quality 85 --progressive

# Imagen 8: dos gruas de 600 ton.png
npx sharp-cli "dos gruas de 600 ton.png" \
-o "optimized/dos-gruas-600-ton.webp" \
--quality 85 --progressive

# Imagen 9: trabajo de grua.png
npx sharp-cli "trabajo de grua.png" \
-o "optimized/trabajo-grua.webp" \
--quality 85 --progressive

# Imagen 10: movilizacion-vagones-ferrocarril.jpg
npx sharp-cli "movilizacion-vagones-ferrocarril.jpg" \
-o "optimized/movilizacion-vagones-ferrocarril.webp" \
--quality 85 --progressive

# 4. Verificar resultados
ls -lh optimized/

```

```
du -sh optimized/
# 5. Si todo se ve bien, mover imágenes al directorio principal
cp optimized/*.webp .
```

Tiempo estimado: 30 minutos

PASO 1.2: Actualizar Referencias en Código

Buscar referencias a imágenes antiguas:

```
cd /home/ubuntu/gruas_equiser_website/app

# Buscar referencias a las 10 imágenes
grep -r "trabajo grua 800 ton.png" components/ app/ --include="*.tsx" --include="*.ts"
grep -r "movilizacion-topas-metro-caracas.png" components/ app/ --include="*.tsx" --include="*.ts"
grep -r "movilizacion-generador-sobredimensionado.png" components/ app/ --include="*.tsx" --include="*.ts"
grep -r "logo-equiser-actualizado.png" components/ app/ --include="*.tsx" --include="*.ts"
grep -r "logo equiser actulizado sin fondo.png" components/ app/ --include="*.tsx" --include="*.ts"
grep -r "trabajo estadio copa america.png" components/ app/ --include="*.tsx" --include="*.ts"
grep -r "trabajo gruas de 600 ton demag.png" components/ app/ --include="*.tsx" --include="*.ts"
grep -r "dos gruas de 600 ton.png" components/ app/ --include="*.tsx" --include="*.ts"
grep -r "trabajo de grua.png" components/ app/ --include="*.tsx" --include="*.ts"
grep -r "movilizacion-vagones-ferrocarril.jpg" components/ app/ --include="*.tsx" --include="*.ts"
```

Reemplazar referencias:

Para cada archivo encontrado, reemplazar manualmente:

```
// ✗ ANTES
<Image src="/images/trabajo grua 800 ton.png" ... />

// ✓ DESPUÉS
<Image src="/images/trabajo-grua-800-ton.webp" ... />
```

O usar sed para reemplazo automático:

```
# Reemplazar en todos los archivos .tsx
find components app -name "*.tsx" -type f -exec sed -i 's|trabajo grua 800 ton\.png|trabajo-grua-800-ton.webp|g' {} +
find components app -name "*.tsx" -type f -exec sed -i 's|movilizacion-topas-metro-caracas\.png|movilizacion-topas-metro-caracas.webp|g' {} +
find components app -name "*.tsx" -type f -exec sed -i 's|movilizacion-generador-sobredimensionado\.png|movilizacion-generador-sobredimensionado.webp|g' {} +
find components app -name "*.tsx" -type f -exec sed -i 's|logo-equiser-actualizado\.png|logo-equiser-actualizado.webp|g' {} +
find components app -name "*.tsx" -type f -exec sed -i 's|logo equiser actualizado sin fondo\.png|logo-equiser-actualizado-sin-fondo.webp|g' {} +
find components app -name "*.tsx" -type f -exec sed -i 's|trabajo estadio copa america\.png|trabajo-estadio-copa-america.webp|g' {} +
find components app -name "*.tsx" -type f -exec sed -i 's|trabajo gruas de 600 ton demag\.png|trabajo-gruas-600-ton-demag.webp|g' {} +
find components app -name "*.tsx" -type f -exec sed -i 's|dos gruas de 600 ton\.png|dos-gruas-600-ton.webp|g' {} +
find components app -name "*.tsx" -type f -exec sed -i 's|trabajo de grua\.png|trabajo-grua.webp|g' {} +
find components app -name "*.tsx" -type f -exec sed -i 's|movilizacion-vagones-ferrocarril\.jpg|movilizacion-vagones-ferrocarril.webp|g' {} +
```

Tiempo estimado: 30 minutos

PASO 1.3: Verificar Calidad Visual

```
# Iniciar servidor de desarrollo
cd /home/ubuntu/gruas_equiser_website/app
yarn dev

# Abrir en navegador
http://localhost:3000
```

Verificar:

1. Home page (hero images)
2. Sección de proyectos
3. Galería de imágenes
4. Blogs individuales
5. Imágenes del footer/header

Checklist:

- Imágenes cargan correctamente
- No hay pérdida visible de calidad
- Velocidad de carga mejorada
- No hay imágenes rotas (404)

Tiempo estimado: 15 minutos

PASO 1.4: Build y Deploy

```
cd /home/ubuntu/gruas_equiser_website/app

# Build
yarn build

# Verificar que build es exitoso
# Debe mostrar: ✓ Compiled successfully
# 176 páginas estáticas generadas

# Deploy
yarn deploy
```

Tiempo estimado: 10 minutos



PASO 2: DECIDIR SOBRE ROUTING DUAL (OPCIONAL)

Problema

Existen DOS sistemas de blogs:

1. **Principal:** /blog/[slug] (105 blogs, Prisma DB) ✓ FUNCIONAL
2. **Secundario:** /[locale]/blog/[slug] (31 blogs, archivos estáticos) ! INACTIVO

Riesgo

Google puede indexar ambas URLs como contenido duplicado.

OPCIÓN A: Eliminar Sistema Secundario (RECOMENDADO)

```
cd /home/ubuntu/gruas_equiser_website/app

# 1. Backup
cp -r app/[locale]/blog app/[locale]/blog.backup

# 2. Eliminar
rm -rf app/[locale]/blog

# 3. Rebuild
yarn build

# 4. Verificar que build es exitoso
# Solo /blog/[slug] debería existir (105 blogs)

# 5. Deploy
yarn deploy
```

Beneficios:

- ✓ Evita contenido duplicado
- ✓ Mejora SEO +15-20%
- ✓ Un solo sistema (simplicidad)

Tiempo estimado: 15 minutos

OPCIÓN B: Redirigir Sistema Secundario (ALTERNATIVA)

```
// Archivo: app/[locale]/blog/[slug]/page.tsx
import { redirect } from 'next/navigation'

export default function BlogPostPage({ params }: { params: { locale: string, slug: string } }) {
  // Redirigir a URL principal sin locale
  redirect(`/${params.slug}`)
}
```

Beneficios:

- Preserva URLs antiguas
- Evita 404s si hay backlinks

Tiempo estimado: 10 minutos

🔍 PASO 3: GOOGLE SEARCH CONSOLE (OPCIONAL PERO RECOMENDADO)

3.1: Enviar Sitemap

1. Ir a Google Search Console:

<https://search.google.com/search-console>

2. Seleccionar propiedad: gruasequierer.com

3. Ir a: Sitemaps

- Menú lateral → Sitemaps

4. Agregar nuevo sitemap:

<https://gruasequierer.com/sitemap.xml>

5. Clic en: Enviar

6. Verificar estado:

- Estado: Éxito
- URLs descubiertas: 107

Tiempo estimado: 5 minutos

3.2: Solicitar Indexación Prioritaria

1. Ir a: Herramienta de inspección de URL

2. Inspeccionar URLs prioritarias:

<https://gruasequierer.com/>
<https://gruasequierer.com/blog>
<https://gruasequierer.com/blog/gruas-moviles-130-toneladas>
<https://gruasequierer.com/blog/costo-alquiler-grua-venezuela>
<https://gruasequierer.com/blog/servicio-gruas-24-7-venezuela>

```
https://gruasequier.com/blog/alquiler-gruas-industriales-venezuela
https://gruasequier.com/blog/transporte-carga-sobredimensionada
https://gruasequier.com/blog/gruas-proyectos-infraestructura-vial
https://gruasequier.com/blog/izamiento-tanques-almacenamiento
https://gruasequier.com/blog/transporte-equipos-mineros
```

3. Para cada URL:

- Clic en: **Solicitar indexación**
- Esperar confirmación (1-2 minutos por URL)

Tiempo estimado: 15-20 minutos



PASO 4: MEDIR RESULTADOS

4.1: PageSpeed Insights (ANTES)

```
# Medir baseline ANTES de optimización

# Mobile
https://pagespeed.web.dev/analysis?url=https://gruasequier.com

# Desktop
https://pagespeed.web.dev/analysis?url=https://gruasequier.com&form_factor=desktop
```

Anotar métricas:

- Mobile Score: 77/100
 - Desktop Score: 94/100
 - LCP: 3.5s
 - FCP: 2.1s
-

4.2: PageSpeed Insights (DESPUÉS)

```
# Medir DESPUÉS de optimización (esperar 5-10 min después de deploy)

# Mobile
https://pagespeed.web.dev/analysis?url=https://gruasequier.com

# Desktop
https://pagespeed.web.dev/analysis?url=https://gruasequier.com&form_factor=desktop
```

Comparar:

- Mobile Score: 77 → 87+ (+10 puntos esperados)
 - Desktop Score: 94 → 98+ (+4 puntos esperados)
 - LCP: 3.5s → <2.5s (-1s)
 - FCP: 2.1s → <1.8s (-0.3s)
-

CHECKLIST FINAL

Tareas Inmediatas (2-3 horas)

- 1. Optimizar TOP 10 imágenes con Squoosh.app o sharp-cli
- 2. Actualizar referencias en código (.tsx/.ts)
- 3. Verificar calidad visual en dev (yarn dev)
- 4. Build y deploy (yarn build && yarn deploy)
- 5. (Opcional) Eliminar sistema de routing dual
- 6. (Opcional) Enviar sitemap a Google Search Console
- 7. (Opcional) Solicitar indexación prioritaria
- 8. Medir PageSpeed Insights (antes y después)

Monitoreo (7-30 días)

- 9. Verificar indexación en Google Search Console
- 10. Monitorear Core Web Vitals
- 11. Analizar tráfico orgánico en Google Analytics
- 12. Verificar posiciones en palabras clave objetivo

RESULTADOS ESPERADOS

Inmediatos (24-48 horas)

-  PageSpeed Mobile: 77 → 87-90 (+10-13 puntos)
-  PageSpeed Desktop: 94 → 98+ (+4 puntos)
-  LCP: 3.5s → <2.5s (29% más rápido)
-  Ahorro bandwidth: 30MB → 5MB (83%)
-  Velocidad de carga: 3x más rápido

Corto Plazo (7-14 días)

-  Indexación completa: 107 páginas
-  Core Web Vitals: 60% → 90%+ "Good"
-  Bounce rate: -15-20%
-  Tiempo en sitio: +30-40%

Medio Plazo (30 días)

-  Tráfico orgánico: +50-100%
-  Impresiones: +30-50%
-  CTR: +10-20%
-  Posición promedio: -5 posiciones (mejora)

PRÓXIMOS PASOS DESPUÉS DE ESTA TAREA

Fase 2: Optimización Adicional (8-10 horas)

1. Optimizar resto de 27 imágenes (500KB-2MB)
2. Implementar srcset responsive (1600px, 1200px, 768px)
3. Configurar caché headers (Browser Cache)
4. Minificar CSS/JS adicional
5. Implementar lazy loading universal

Fase 3: Contenido y Backlinks (continuo)

1. Publicar 2-3 blogs nuevos por semana
2. Actualizar blogs existentes (**freshness**)
3. Estrategia de backlinks (directorios, guest **posting**)
4. Social media marketing
5. **Google My Business** optimización

Fecha de creación: 15 de diciembre, 2025

Estado:  **LISTO PARA EJECUTAR**

Próxima acción: Optimizar TOP 10 imágenes

 **EMPEZAR AHORA: OPTIMIZAR IMÁGENES CRÍTICAS**