

Informe Integral de Auditoría Técnica y Estrategia de Remedición para GrupoTranservica.com: Optimización de Rendimiento, Accesibilidad y Cumplimiento SEO en Arquitectura React/Vite

1. Resumen Ejecutivo y Diagnóstico Arquitectónico

En el ecosistema actual de desarrollo web, la convergencia entre el rendimiento de carga (Performance), la accesibilidad universal (A11y) y la descubribilidad en motores de búsqueda (SEO) constituye el estándar operativo mínimo para cualquier activo digital corporativo. Para grupotranservica.com, una aplicación de página única (SPA) construida sobre el stack tecnológico de React y Vite y alojada en la infraestructura de Replit, el desafío de alcanzar una puntuación perfecta de 100 en las auditorías de Lighthouse y resolver los errores críticos de Google Search Console no es trivial. Requiere una reingeniería profunda de la lógica de servicio estático y una optimización granular de los recursos del lado del cliente.

El diagnóstico actual revela una disonancia fundamental entre la arquitectura de enrutamiento del lado del cliente (Client-Side Routing) y la configuración del servidor de archivos estáticos de Replit. Esta desalineación es la causa raíz de los errores 404 en rutas profundas y, consecuentemente, del estado "Rastreada - actualmente no indexada" en Search Console. Paralelamente, el rendimiento de la aplicación se ve comprometido por métricas deficientes en el Despliegue de Contenido Más Grande (LCP) y el Tiempo Total de Bloqueo (TBT), exacerbados por la inyección ineficiente de iframes de YouTube y activos de imagen no optimizados. Desde la perspectiva de la accesibilidad, la paleta de colores corporativa actual infringe los estándares WCAG 2.1 AA, y los componentes interactivos carecen de la semántica necesaria para las tecnologías de asistencia.

Este informe técnico proporciona una hoja de ruta exhaustiva y detallada para dismantelar sistemáticamente estos cuellos de botella. La estrategia prioriza la estabilización de la infraestructura de enrutamiento mediante reglas de reescritura en el servidor, seguida de la implementación de patrones de "fachada" para la carga diferida de medios, y culmina con la integración de tuberías de generación de mapas de sitio dinámicos y cabeceras de seguridad estrictas. La ejecución de este plan transformará grupotranservica.com de una aplicación frágil en una plataforma robusta, segura y altamente performante.

2. Infraestructura de Enrutamiento y Resolución de Errores de Indexación

La integridad estructural de una SPA depende de la capacidad del servidor para delegar el control de la navegación al cliente sin interrumpir el flujo de peticiones HTTP esperadas por los rastreadores de búsqueda. La actual configuración de grupotranservica.com en Replit presenta fallos críticos en este traspaso de control.

2.1. Mecánica de Fallos de Enrutamiento SPA en Alojamientos Estáticos

Para comprender la naturaleza del error 404 que afecta a las subpáginas y su impacto en la indexación, es necesario analizar el ciclo de vida de una petición HTTP en este entorno. En una aplicación React estándar que utiliza react-router-dom, la lógica de enrutamiento se ejecuta exclusivamente en el navegador del usuario. Cuando un usuario navega desde la página de inicio (/) hacia una sección interna como /servicios, el router de React intercepta la acción y actualiza la vista manipulando el DOM y el historial del navegador mediante la API de Historia de HTML5, todo ello sin realizar una nueva solicitud al servidor.

Sin embargo, el escenario cambia drásticamente cuando la navegación no es iniciada por el usuario desde la aplicación, sino que proviene de una fuente externa: un enlace directo, una actualización de la página (F5) o, crucialmente, el rastreo de un bot de Google. En este caso, el navegador o el bot envía una solicitud HTTP GET al servidor de Replit solicitando el recurso /servicios.

Los servidores estáticos, por diseño, operan bajo una lógica de resolución de archivos literal. Al recibir la petición para /servicios, el servidor busca en su directorio raíz un archivo físico llamado servicios.html. Dado que el proceso de construcción (build) de Vite para una SPA genera típicamente un único punto de entrada (index.html) y un paquete de activos JavaScript/CSS en la carpeta dist, el archivo servicios.html no existe. En consecuencia, el servidor responde correctamente —desde su lógica interna— con un código de estado 404 Not Found.¹

Esta respuesta 404 es devastadora para el SEO. Incluso si el servidor está configurado para mostrar una página de error personalizada que eventualmente carga la aplicación y redirige al usuario, el encabezado HTTP inicial ya ha comunicado a Googlebot que la página no existe. Los rastreadores de búsqueda son, en gran medida, agentes sin estado que toman decisiones de indexación basadas en los códigos de respuesta iniciales. Un 404 indica que el contenido es efímero o erróneo, provocando que la URL sea descartada del índice o categorizada bajo el estado "Rastreada - actualmente no indexada".⁴

2.2. Implementación de Reglas de Reescritura (Rewrites) en Replit

La solución técnica no reside en modificar el código de React, sino en reconfigurar el comportamiento del servidor de despliegue estático de Replit. El objetivo es transformar el servidor en un servidor de aplicaciones de página única mediante la implementación de una regla de reescritura "catch-all" (atrapalo-todo).

Esta configuración instruye al servidor para que, ante cualquier solicitud de una ruta que no corresponda a un archivo físico existente, sirva el archivo `index.html` pero mantenga la URL original en la barra de direcciones y devuelva un código de estado 200 OK. Esto permite que el paquete de JavaScript de la aplicación se cargue, lea la URL actual (`window.location.pathname`), e hidrate el componente React correspondiente a esa ruta.

Directiva de Configuración en `.replit`:

La plataforma Replit utiliza el archivo `.replit` en la raíz del proyecto para gestionar la configuración del entorno de despliegue. Para solucionar los errores 404 y los problemas de indexación, se debe insertar la siguiente directiva de reescritura:

Ini, TOML

```
[[deployment.rewrites]]
from = "/*"
to = "/index.html"
```

Esta instrucción es precisa y potente. El parámetro `from = "/*"` actúa como un comodín que coincide con cualquier ruta solicitada al dominio. El parámetro `to = "/index.html"` define el recurso que se servirá en respuesta.

El Fenómeno del "Shadowing" (Sombreado):

Una preocupación técnica válida al implementar una regla de reescritura global es el riesgo de interceptar solicitudes legítimas de activos estáticos, como imágenes, hojas de estilo CSS o scripts JS. Si la regla `/*` redirigiera *todo* al `index.html`, la aplicación se rompería porque el navegador recibiría el contenido HTML del `index` en lugar del código binario de una imagen o el texto de un script.

Afortunadamente, la infraestructura de despliegue estático de Replit implementa un comportamiento de resolución denominado "Shadowing". Bajo este modelo, el servidor verifica prioritariamente la existencia física del archivo solicitado. Si un usuario solicita `/assets/logo.png` y este archivo existe en el directorio de publicación, el servidor lo entrega inmediatamente, ignorando la regla de reescritura. La regla de reescritura solo se activa ("shadows") cuando no se encuentra un archivo coincidente en el sistema de archivos. Esto

garantiza que la arquitectura de activos de Vite funcione sin interrupciones mientras se solucionan los problemas de enrutamiento.⁶

2.3. Estrategia de Corrección para "Rastreada - Actualmente No Indexada"

El estado "Rastreada - actualmente no indexada" en Google Search Console es un indicador matizado. Significa que Google ha descubierto la URL y ha enviado al Googlebot para analizarla, pero ha tomado la decisión algorítmica de no incluirla en el índice. Una vez resuelto el problema técnico del código 404 mediante las reescrituras, persisten otros factores que pueden mantener este estado y que deben ser abordados.

Análisis de la Cadena de Renderizado:

Googlebot posee capacidades de renderizado de JavaScript, pero estas no son infinitas ni inmediatas. Si grupotranservica.com sirve una "cáscara vacía" (un div vacío con id root) y depende de un paquete JavaScript pesado para renderizar el contenido, existe el riesgo de que el tiempo de ejecución del bot se agote antes de que el contenido significativo aparezca. Esto es especialmente crítico si hay bloqueos en el hilo principal (TBT) causados por scripts de terceros.

Soluciones Integrales:

1. **Validación de Código de Estado:** Tras aplicar la configuración en .replit, es imperativo utilizar la herramienta de "Inspección de URL" en Search Console para confirmar que las rutas profundas (ej. /blog/articulo-1) devuelven un estado HTTP 200.
2. **Metaetiquetas Canónicas Dinámicas:** Una causa frecuente de no indexación en SPAs es la duplicidad de contenido percibida. Si la aplicación es accesible tanto por <https://grupotranservica.com/home> como por <https://grupotranservica.com/>, Google puede "confundirse". Se debe implementar una lógica canónica estricta que se autorreferencie en cada vista dinámica.⁷
3. **Enlazado Interno y Mapas de Sitio:** Las páginas huérfanas (sin enlaces entrantes) son candidatas principales para no ser indexadas. La generación de un sitemap XML dinámico que incluya todas las rutas de blog y servicios es crucial para declarar explícitamente la estructura del sitio al motor de búsqueda.⁹

3. Ingeniería de Rendimiento Web: Optimización de Core Web Vitals

El rendimiento web en 2026 se mide a través de las Core Web Vitals de Google. Para grupotranservica.com, el análisis identifica dos áreas críticas que impiden alcanzar el puntaje de 100 en Lighthouse: el Despliegue del Contenido Más Grande (LCP) y el Tiempo Total de Bloqueo (TBT).

3.1. Eliminación del Bloqueo del Hilo Principal mediante Fachadas de Medios

La inclusión de videos de YouTube mediante etiquetas <iframe> estándar es una de las prácticas más perjudiciales para el rendimiento inicial de una página. Al cargar la página, un iframe de YouTube descarga inmediatamente la biblioteca completa del reproductor (aproximadamente 500KB - 1.2MB de JavaScript comprimido), inicia múltiples conexiones SSL a dominios de Google y comienza el análisis de scripts. Todo esto ocurre en el hilo principal del navegador, compitiendo con la hidratación de la aplicación React y provocando un aumento masivo en el Tiempo Total de Bloqueo (TBT).¹⁰

Para una aplicación que busca la excelencia técnica, este comportamiento es inaceptable. La solución arquitectónica es la implementación del patrón de "Fachada" (Facade Pattern).

Implementación Técnica con react-lite-youtube-embed:

Este enfoque sustituye el iframe pesado por un componente ligero que simula visualmente el reproductor de video. Inicialmente, solo se carga la imagen en miniatura (thumbnail) del video y un botón de reproducción estilizado con CSS. El costo en JavaScript es insignificante (menos de 5KB). El recurso real de YouTube solo se invoca e inyecta en el DOM cuando el usuario interactúa explícitamente con el elemento (hace clic en "Play").

Directrices de Implementación para React/Vite:

Se recomienda la biblioteca react-lite-youtube-embed por su eficiencia y soporte de tipos. La implementación debe maximizar el rendimiento mediante el uso de propiedades específicas:

1. **poster="maxresdefault"**: Obliga a cargar la miniatura de mayor resolución disponible. Esto es crucial para evitar el Cambio Acumulativo de Diseño (CLS) que ocurriría si una miniatura de baja calidad tiene dimensiones intrínsecas diferentes al contenedor del video.
2. **lazyLoad={true}**: Aplica carga diferida nativa (loading="lazy") a la imagen de la miniatura, asegurando que no consuma ancho de banda si el video está fuera de la ventana gráfica inicial (viewport).¹²
3. **webp={true}**: Solicita a los servidores de YouTube la versión WebP de la miniatura, reduciendo el tamaño del archivo de imagen en un 30-50% comparado con JPG.

TypeScript

```
// Ejemplo de componente optimizado en React
import LiteYouTubeEmbed from 'react-lite-youtube-embed';
import 'react-lite-youtube-embed/dist/LiteYouTubeEmbed.css';
```

```
export const OptimizedVideo = ({ videoId, title }: { videoId: string, title: string }) => {
  return (
    <div className="aspect-w-16 aspect-h-9 rounded-lg overflow-hidden shadow-xl">
      <LiteYouTubeEmbed
        id={videoId}
        title={title}
        poster="maxresdefault"
        lazyLoad={true}
        webp={true}
      />
    </div>
  );
};
```

La adopción de esta estrategia elimina prácticamente todo el impacto negativo de los videos en el TBT y el Tiempo de Interactividad (TTI), factores de gran peso en la puntuación global de Lighthouse.¹³

3.2. Optimización de LCP mediante Procesamiento Avanzado de Imágenes

El Despliegue del Contenido Más Grande (LCP) mide cuándo el elemento principal de la página (generalmente una imagen de héroe o un bloque de texto) se vuelve visible. En el caso analizado, se detectó un logotipo de dimensiones masivas (1738x1482 píxeles) siendo renderizado en un espacio de apenas 128 píxeles de ancho.

Análisis del Impacto:

Servir una imagen de 1700px para un contenedor de 128px es un error de rendimiento crítico. El navegador debe descargar megabytes de datos innecesarios (retrasando el LCP) y luego utilizar ciclos de CPU intensivos para escalar la imagen a su tamaño de visualización. Además, si las dimensiones no están explícitamente declaradas en el HTML, el navegador no puede reservar el espacio antes de la carga, provocando cambios de diseño (CLS).

Tubería de Optimización Automatizada en Vite:

Para resolver esto de manera escalable y sostenible, no se deben redimensionar las imágenes manualmente una a una. Se debe configurar una tubería de optimización en el proceso de construcción (build) de Vite utilizando dos complementos esenciales: vite-plugin-image-optimizer y vite-imagemeta.

Configuración de vite.config.js:

JavaScript

```
import { defineConfig } from 'vite';
import { VitelImageOptimizer } from 'vite-plugin-image-optimizer';
import { imagetools } from 'vite-imagetools';

export default defineConfig({
  plugins: [
    // Optimización sin pérdidas para todos los activos
    VitelImageOptimizer({
      png: { quality: 80 },
      jpeg: { quality: 75 },
      webp: { lossless: true },
      avif: { lossless: true },
    }),
    // Herramientas para generación de variantes responsivas
    imagetools(),
  ],
});
```

Implementación de Imágenes Responsivas:

Utilizando vite-imagetools, podemos importar la imagen y solicitar variantes específicas directamente en la sentencia de importación. Esto genera múltiples versiones optimizadas en tiempo de compilación.

JavaScript

```
// Importación que genera tres variantes WebP de diferentes anchos
import logoSrcset from './assets/logo.png?w=128;256;512&format=webp&as=srcset';
import logoFallback from './assets/logo.png?w=256&format=jpg';

// Uso en JSX con atributo sizes para la selección del navegador
<img
  srcSet={logoSrcset}
  src={logoFallback}
  width="128"
```

```
height="109"  
sizes="(max-width: 600px) 128px, 256px"  
alt="Logotipo de Grupo Transservica"  
className="logo-class"  
</>
```

Corrección de Relación de Aspecto (Aspect Ratio): Para solucionar la alerta de Lighthouse sobre "Image elements do not have explicit width and height", es imperativo declarar los atributos width y height en la etiqueta . Estos valores deben reflejar la relación de aspecto intrínseca de la imagen renderizada (128x109 en este ejemplo). Esto permite al navegador calcular la relación de aspecto y reservar el espacio en blanco en el diseño antes de que se descargue el primer byte de la imagen, eliminando el CLS y mejorando la estabilidad visual percibida.¹⁴

4. Estrategia de SEO Técnico: Mapas de Sitio y Metadatos Dinámicos

La infraestructura técnica de una SPA debe ser complementada con señales explícitas para los motores de búsqueda. La "descubribilidad" no puede dejarse al azar del rastreo de enlaces JavaScript.

4.1. Generación de Sitemaps para Rutas Dinámicas

Los generadores de mapas de sitio estáticos a menudo fallan en aplicaciones React porque no pueden inferir las rutas que dependen de datos de una API (como /blog/articulo-1, /blog/articulo-2). La herramienta vite-plugin-sitemap debe configurarse para inyectar estas rutas dinámicas durante el tiempo de construcción.

Estrategia de Configuración:

El archivo vite.config.js debe incluir lógica para definir estas rutas. En un entorno de producción real, esto podría implicar una llamada a la API del CMS durante el proceso de build para obtener la lista de slugs actuales.

JavaScript

```
// vite.config.js  
import Sitemap from 'vite-plugin-sitemap';  
  
// Lista simulada o obtenida de rutas dinámicas  
const blogRoutes = [
```



```

    '/blog/optimizacion-flotas',
    '/blog/logistica-internacional',
    '/servicios/transporte-carga'
  ];

export default defineConfig({
  plugins:
  })
}
});

```

Esta configuración genera automáticamente los archivos sitemap.xml y robots.txt en la carpeta dist. Al subir estos archivos al servidor, se proporciona a Googlebot un mapa completo y priorizado de la estructura del sitio, facilitando la indexación de páginas profundas que de otro modo quedarían ocultas tras la interacción del usuario.¹⁷

4.2. Gestión de Metadatos con React Helmet Async

Para resolver los errores de "Duplicidad" o "Página alternativa con etiqueta canónica adecuada" en Search Console, cada vista de la aplicación debe inyectar metadatos únicos y, lo más importante, una etiqueta canónica autorreferencial precisa. Se debe utilizar react-helmet-async en lugar del obsoleto react-helmet, ya que ofrece mejor soporte para el asincronismo y la seguridad de hilos en renderizado moderno.

Lógica de Canonicalización:

El componente Helmet debe configurarse para normalizar la URL actual, eliminando parámetros de consulta no esenciales (como fbclid o utm_source) que diluyen la autoridad de la página.

JavaScript

```

import { Helmet } from 'react-helmet-async';
import { useLocation } from 'react-router-dom';

const SEO = ({ title, description }) => {
  const location = useLocation();
  // Limpieza de parámetros de consulta para la URL canónica
  const canonicalUrl = `https://grupotranservica.com${location.pathname.split('?')}`;

```

```

return (
  <Helmet>
    <title>`${title} | Grupo Transservica`</title>
    <meta name="description" content={description} />
    <link rel="canonical" href={canonicalUrl} />
    <meta property="og:title" content={title} />
    <meta property="og:url" content={canonicalUrl} />
    <meta name="robots" content="index, follow" />
  </Helmet>
);
};

```

Esta implementación asegura que, independientemente de cómo un usuario llegue a una página (por ejemplo, a través de una campaña de marketing con parámetros de seguimiento), Google siempre reciba la señal de que la versión "limpia" de la URL es la fuente autoritativa, consolidando la equidad de enlaces y resolviendo conflictos de indexación.⁷

5. Cumplimiento de Accesibilidad (WCAG 2.1 AA)

La accesibilidad web trasciende el cumplimiento normativo; es un indicador de calidad de ingeniería. Las auditorías revelaron violaciones específicas en el contraste de color y la navegación móvil.

5.1. Remedición Algorítmica del Contraste de Color

El color verde corporativo identificado, #3ea30f, sobre un fondo blanco (#FFFFFF), presenta una relación de contraste de luminosidad de aproximadamente **2.97:1**. Esto incumple el estándar WCAG 2.1 AA, que exige un mínimo de **4.5:1** para texto normal y **3.0:1** para texto grande o componentes de interfaz gráfica (UI).²⁰

Análisis Colorimétrico y Solución:

Para mantener la identidad de marca mientras se cumple con la normativa, es necesario reducir la luminancia del color (oscurecerlo) hasta alcanzar el umbral matemático.

- **Color Actual (Fallo):** #3ea30f (Ratio: ~2.97:1)
- **Alternativa Cumplimiento AA:** #2d7a0b (Ratio: ~4.52:1). Este tono es perceptualmente similar pero matemáticamente conforme.
- **Alternativa Óptima (Mejor Legibilidad):** #246109 (Ratio: ~5.80:1). Ofrece un margen de seguridad mayor para usuarios con baja visión o pantallas con brillo reducido.

Recomendación de Implementación: Se debe actualizar la configuración del tema (ej. tailwind.config.js o variables CSS :root) para reemplazar globalmente el color de texto e iconos interactivos. El color original #3ea30f puede mantenerse únicamente para elementos

decorativos grandes que no contengan información textual crítica.²²

5.2. Accesibilidad Móvil en Carruseles (SwiperJS)

Los carruseles (sliders) son componentes complejos para la accesibilidad. En la versión móvil, los botones de navegación de SwiperJS a menudo se renderizan como elementos div o span sin etiquetas semánticas, lo que los hace invisibles o confusos para los lectores de pantalla.

Configuración del Módulo A11y:

SwiperJS incluye un módulo de accesibilidad (A11y) que debe habilitarse explícitamente. Este módulo gestiona atributos ARIA dinámicos (como aria-live para anunciar cambios de diapositiva) y permite definir etiquetas legibles para los controles.

JavaScript

```
import { Navigation, Pagination, A11y } from 'swiper/modules';
import { Swiper, SwiperSlide } from 'swiper/react';
```

```
<Swiper
  modules={[Navigation, Pagination, A11y]}
  a11y={{
    enabled: true,
    prevSlideMessage: 'Diapositiva anterior',
    nextSlideMessage: 'Siguiente diapositiva',
    firstSlideMessage: 'Esta es la primera diapositiva',
    lastSlideMessage: 'Esta es la última diapositiva',
    paginationBulletMessage: 'Ir a la diapositiva {{index}}',
  }}
  navigation={true}
>
  { /* Diapositivas */ }
</Swiper>
```

Esta configuración transforma la experiencia de un usuario invidente. En lugar de escuchar "botón no etiquetado", el lector de pantalla anunciará claramente la función del control ("Siguiente diapositiva"), cumpliendo con los criterios de éxito de navegación y operatividad de WCAG.²⁴

6. Seguridad, Despliegue y Endurecimiento de la

Infraestructura

La profesionalización del despliegue en Replit implica asegurar la aplicación contra ataques comunes y gestionar adecuadamente la separación de entornos.

6.1. Política de Seguridad de Contenidos (CSP) y Encabezados HTTP

La implementación de encabezados de seguridad robustos es una defensa proactiva contra ataques XSS (Cross-Site Scripting) e inyección de datos. En Replit, esto se configura mediante la directiva `[[deployment.responseHeaders]]` en el archivo `.replit`.

Configuración de CSP para Recursos Externos:

Dado que la aplicación utiliza iframes de YouTube y fuentes de Google Fonts, la Política de Seguridad de Contenidos (CSP) debe incluir excepciones específicas ("allowlists") para estos dominios, bloqueando todo lo demás por defecto.

Ini, TOML

```
# Configuración de seguridad en.replit
[[deployment.responseHeaders]]
path = "/*"
name = "Content-Security-Policy"
value = "default-src 'self'; script-src 'self' 'unsafe-inline' https://www.youtube.com https://s.ytimg.com;
frame-src https://www.youtube.com; style-src 'self' 'unsafe-inline' https://fonts.googleapis.com;
font-src 'self' https://fonts.gstatic.com; img-src 'self' data: https://i.ytimg.com;"
```

```
[[deployment.responseHeaders]]
path = "/*"
name = "X-Content-Type-Options"
value = "nosniff"
```

```
[[deployment.responseHeaders]]
path = "/*"
name = "Strict-Transport-Security"
value = "max-age=31536000; includeSubDomains"
```

Política de Permisos (Permissions-Policy) y Hardware:

Lighthouse suele alertar sobre violaciones de la política de permisos relacionadas con el acelerómetro y el giroscopio en iframes de video. Para solucionar esto sin comprometer la

funcionalidad, se debe asegurar que el componente LiteYouTubeEmbed (o cualquier iframe residual) incluya el atributo allow con los permisos necesarios delegados explícitamente.

Atributo recomendado para el iframe/componente:

```
allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"
```

Esto delega el permiso solo al contexto del video, satisfaciendo la auditoría de seguridad del navegador y de Lighthouse.²⁵

6.2. Gestión de Entornos y Marca "Made with Replit"

Eliminación de la Marca de Replit: Es importante aclarar técnicamente la situación del banner "Made with Replit". No existe una variable de configuración o "flag" oficial en el código para desactivar esto en los planes gratuitos o básicos. La documentación y la comunidad confirman que la eliminación oficial está vinculada al nivel de servicio (Planes Core/Teams) y al uso de **Dominios Personalizados**. Al vincular grupotranservica.com (como se indica en el objetivo), la superposición de marca de Replit suele minimizarse o desaparecer, ya que el dominio personalizado indica un despliegue de producción. Intentar ocultarlo mediante inyección de CSS (ej. display: none sobre la clase del badge) es técnicamente posible pero frágil y desaconsejado por violar los términos de servicio.²⁷

Segregación de Entornos (Dev vs Prod):

Para evitar que características de desarrollo (como herramientas de depuración o banners de prueba) se filtren a producción, se deben utilizar las **Secrets** (Secretos) de Replit como variables de entorno.

1. Definir un secreto en Replit: VITE_APP_ENV = production.
2. En el código React, condicionar la lógica:
JavaScript

```
if (import.meta.env.VITE_APP_ENV !== 'production') {  
  // Mostrar herramientas de desarrollo  
}
```
3. Replit permite mantener secretos "desincronizados" deliberadamente entre el entorno de edición y el despliegue, lo cual es la práctica correcta para aislar configuraciones.²⁹

7. Lista de Verificación de Implementación y Conclusiones

Para materializar la optimización al 100%, el equipo de ingeniería debe ejecutar las siguientes operaciones en el orden prescrito. Este flujo asegura que las correcciones estructurales

(enrutamiento) precedan a las optimizaciones de contenido.

Fase	Categoría	Acción Técnica	Detalle de Implementación	Impacto Directo
1	Infraestructura	Reglas de Reescritura	Añadir <code>[[deployment.rewrites]]</code> en <code>.replit</code> mapeando <code>/*</code> a <code>/index.html</code> .	Resuelve errores 404 y permite indexación.
2	SEO Técnico	Sitemap Dinámico	Configurar vite-plugin-sitemap inyectando rutas de blog y servicios.	Soluciona "Descubierta - no indexada".
3	SEO On-Page	Canonicalización	Implementar componente <code><SEO /></code> con <code>react-helmet-async</code> en todas las rutas.	Previene contenido duplicado.
4	Rendimiento	Fachada YouTube	Reemplazar iframes por <code>react-lite-youtube-embed</code> con carga diferida.	Reduce drásticamente TBT y mejora LCP.
5	Rendimiento	Activos de Imagen	Integrar vite-plugin-image-optimizer y definir <code>width/height</code> explícitos.	Elimina CLS y optimiza la carga de red.

6	Accesibilidad	Corrección de Color	Modificar variables CSS de verde #3ea30f a #2d7a0b.	Cumplimiento WCAG 2.1 AA (Contraste).
7	Accesibilidad	Semántica UI	Activar módulo A11y en SwiperJS con etiquetas ARIA localizadas.	Navegación accesible para lectores de pantalla.
8	Seguridad	Cabeceras CSP	Definir políticas estrictas en .replit permitiendo solo orígenes confiables.	Protección XSS y cumplimiento de mejores prácticas.

Conclusión:

Los desafíos actuales de grupotranservica.com no son indicativos de un código defectuoso, sino de una desalineación de configuración entre la arquitectura moderna de SPA y el entorno de servicio estático. Al implementar la reescritura de rutas en el servidor, la aplicación ganará visibilidad total ante los motores de búsqueda. Simultáneamente, la transición hacia patrones de fachada para medios y la optimización de activos reducirán la carga de procesamiento en los dispositivos de los clientes, traducándose directamente en puntuaciones de Core Web Vitals superiores y una experiencia de usuario de primer nivel. Finalmente, el compromiso con los estándares de contraste y semántica WCAG asegurará que el sitio sea robusto, inclusivo y legalmente conforme.

Aquí tienes un ****informe técnico completo de optimización**** para llevar tu sitio web de TRANSERVICA al 100% en rendimiento, con enfoque especial en dispositivos móviles y velocidad máxima.

📊 INFORME DE OPTIMIZACIÓN WEB - TRANSERVICA

Objetivo: Rendimiento 100% | Mobile-First | Carga Instantánea

🎯 RESUMEN EJECUTIVO

Métrica	Actual Mobile	Actual Desktop	Objetivo
-----	-----	-----	-----
Performance	56	71	**100**
FCP	2.8s	0.7s	<1.8s
LCP	6.6s	2.1s	<2.5s
TBT	440ms	400ms	<200ms
CLS	0	0.018	0
SI	6.0s	1.5s	<3.4s

****Problema crítico:**** El JavaScript de YouTube está destruyendo el rendimiento (1.5MB+ de scripts innecesarios).

🚨 PRIORIDAD 1: CRÍTICO (Impacto Inmediato)

1.1 Eliminar/Optimizar YouTube (Ahorro: ~1.5MB, ~900ms TBT)

****Problema:**** Los iframes de YouTube cargan 1.5MB de JavaScript bloqueante.

****Soluciones:****

```
```html
```

```
<!-- OPCIÓN A: Lazy Load con thumbnail personalizado -->
```

```
<div class="youtube-embed" data-video-id="_LQbWkWlg6s">
```

```

```

```
<button class="play-button" aria-label="Reproducir video">
```

```
 <svg><!-- Icono play --></svg>
```

```
</button>
```

```
</div>
```

```
<script>
```

```
// Cargar iframe solo al hacer clic
```

```

document.querySelectorAll('.youtube-embed').forEach(el => {

 el.addEventListener('click', () => {

 const id = el.dataset.videoid;

 el.innerHTML = `<iframe

 src="https://www.youtube.com/embed/${id}?autoplay=1"

 loading="lazy"

 width="640"

 height="360"

 allowfullscreen

 ></iframe>`;

 });

});

</script>

...

```

```

```css

```

```

/* OPCIÓN B: Facade con lite-youtube-embed */

```

```

@import 'lite-youtube-embed/src/lite-yt-embed.css';

```

```

<lite-youtube videoid="_LQbWkWlg6s" style="background-image:
url('/assets/thumbnails/video-1.webp');">

```

```

<button type="button" class="lty-playbtn">

```

```

  <span class="lyt-visually-hidden">Reproducir</span>

```

</button>

</lite-youtube>

...

****Impacto esperado:**** Reducción de 1.4s en LCP, TBT < 200ms.

1.2 Optimización de Imágenes (Ahorro: 323KB móvil, 310KB desktop)

****Problema actual:****

- Logo: 1738x1482px mostrándose a 197x168px (móvil) / 150x128px (desktop)
- Imagen proyecto: 1920x1081px mostrándose a 665x499px

****Solución técnica:****

```html

<!-- Logo con srcset responsive -->


```

<!-- Imagen de proyecto con lazy loading nativo -->

```
<picture>  
  <source  
    srcset="/assets/proyecto-400.webp 400w,  
            /assets/proyecto-800.webp 800w,  
            /assets/proyecto-1200.webp 1200w"  
    sizes="(max-width: 768px) 100vw, 50vw"  
    type="image/webp"  
  >
```

```


</picture>

...
```

****Comandos de optimización (ImageMagick):****

```
```bash
```

```
Generar versiones optimizadas
```

```
convert logo-original.png -resize 200x -quality 85 -define webp:target-size=15kb
logo-200.webp
```

```
convert logo-original.png -resize 400x -quality 85 logo-400.webp
```

```
Para fotos
```

```
cwebp -q 80 -size 50k proyecto-original.jpg -o proyecto-400.webp
```

```
...
```

```

```

**### 1.3 Eliminar JavaScript No Utilizado (Ahorro: 37.6KB propio + 968KB YouTube)**

**\*\*Análisis:\*\*** Tu bundle principal tiene código muerto.

**\*\*Solución - Code Splitting:\*\***

```
````javascript
// vite.config.js o webpack

export default {

  build: {

    rollupOptions: {

      output: {

        manualChunks: {

          // Separar por rutas/funcionalidad

          'home': ['./src/pages/Home.jsx'],

          'contact': ['./src/pages/Contact.jsx'],

          'animations': ['./src/utls/animations.js'],

        }

      }

    },

    // Eliminar console.logs en producción

    terserOptions: {

      compress: {

        drop_console: true,

        drop_debugger: true

      }

    }

  }

}
```

```
    }  
  }  
}  
}  
...  
  
---
```

****Lazy loading de componentes:****

```
```javascript
```

```
// En lugar de import estático
```

```
import ContactSection from './ContactSection';
```

```
// Usar React.lazy
```

```
const ContactSection = lazy(() => import('./ContactSection'));
```

```
const EquipmentCarousel = lazy(() => import('./EquipmentCarousel'));
```

```
// Suspense con fallback optimizado
```

```
<Suspense fallback={<div class="skeleton" style={{height: '400px'}} />}>
```

```
 <ContactSection />
```

```
</Suspense>
```

```
...

```

## ## 🟡 PRIORIDAD 2: ALTO (Mejoras Significativas)

### ### 2.1 Optimizar CSS Crítico (Ahorro: 13-14KB)

**\*\*Problema:\*\*** 18KB de CSS, 13KB no se usan en la vista inicial.

**\*\*Solución - Critical CSS:\*\***

```
```html
<head>

  <!-- CSS crítico inline (< 14KB para que quepa en el primer paquete TCP) -->

  <style>

    /* Solo estilos above-the-fold */

    :root{--color-primary:#155d29;--color-secondary:#3ea30f}

    *,*::before,*::after{box-sizing:border-box;margin:0}

    body{font-family:system-ui,-apple-system,sans-serif;line-height:1.5}

    .hero{min-height:100vh;display:flex;align-items:center}

    /* ... más estilos críticos ... */

  </style>

  <!-- CSS no crítico cargado asíncronamente -->

  <link rel="preload" href="/assets/critical.css" as="style"
  onload="this.onload=null;this.rel='stylesheet'">
```



```
<noscript><link rel="stylesheet" href="/assets/critical.css"></noscript>
```

```
<!-- Resto de CSS diferido -->
```

```
<link rel="preload" href="/assets/async.css" as="style" onload="this.rel='stylesheet'">
```

```
</head>
```

```
...
```

2.2 Preconnect y DNS-Prefetch

```
```html
```

```
<head>
```

```
<!-- Establecer conexiones anticipadas -->
```

```
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

```
<link rel="dns-prefetch" href="https://www.youtube.com">
```

```
<link rel="dns-prefetch" href="https://static.doubleclick.net">
```

```
<!-- Precargar fuente crítica -->
```

```
<link rel="preload" href="/fonts/inter-400.woff2" as="font" type="font/woff2" crossorigin>
```

```
</head>
```

```
...
```

### ### 2.3 Service Worker para Caching Estratégico

```
````javascript

// sw.js - Cache First Strategy

const CACHE_NAME = 'transervica-v1';

const STATIC_ASSETS = [

  '/',

  '/assets/logo-200.webp',

  '/assets/index.css',

  '/assets/index.js'

];

self.addEventListener('install', (e) => {

  e.waitUntil(

    caches.open(CACHE_NAME)

      .then(cache => cache.addAll(STATIC_ASSETS))

  );

});

self.addEventListener('fetch', (e) => {

  e.respondWith(

    caches.match(e.request)

      .then(response => response || fetch(e.request))

  );

});
```

```

---

## 🟢 PRIORIDAD 3: MEDIO (Polish & Accesibilidad)

### 3.1 Accesibilidad - Botones sin nombre

**\*\*Problema:\*\*** Los botones del carrusel no tienen texto accesible.

**\*\*Fix inmediato:\*\***

```jsx

// Antes (problema)

```
<button className="w-3 h-3 rounded-full bg-gray-300" />
```

// Después (solución)

```
<button
```

```
  className="w-3 h-3 rounded-full bg-gray-300"
```

```
  aria-label={`Ir a proyecto ${index + 1}`}
```

```
  aria-current={isActive ? 'true' : 'false'}
```

```
>
```

```
  <span className="sr-only">Proyecto {index + 1}</span>
```

```
</button>
```

...

3.2 Contraste de colores

****Problema:**** `#3ea30f` sobre blanco tiene ratio 2.9:1 (necesita 4.5:1).

****Solución:****

```css

/\* Reemplazar por tono más oscuro \*/

.bg-transervica-green { background-color: #155d29; } /\* 7.2:1 ✓ \*/

.text-badge {

background-color: #2d7a0d; /\* 4.6:1 ✓ \*/

color: white;

}

...

### ### 3.3 CLS (Cumulative Layout Shift)

**\*\*Problema:\*\*** Las fuentes de Google causan CLS de 0.018.

**\*\*Solución - Font Display:\*\***

```html

<link

href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap"

```
rel="stylesheet">
```

```
...
```

```
```css
```

```
/* Fallback font para reducir CLS */
```

```
@font-face {
```

```
 font-family: 'Inter';
```

```
 font-style: normal;
```

```
 font-weight: 400;
```

```
 font-display: swap; /* ¡Crítico! */
```

```
 src: url('/fonts/inter-400.woff2') format('woff2');
```

```
}
```

```
/* Tamaños de fuente explícitos para evitar layout shifts */
```

```
body {
```

```
 font-family: 'Inter', system-ui, -apple-system, sans-serif;
```

```
 font-size: 16px;
```

```
 line-height: 1.5;
```

```
}
```

```
...
```

```

```

## ## 📱 OPTIMIZACIONES ESPECÍFICAS MÓVIL

### ### 4.1 Viewport y Touch Targets

```
```html
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0,  
maximum-scale=5.0">
```

```
```
```

```
```css
```

```
/* Touch targets mínimos 48x48px */
```

```
button, a {
```

```
    min-height: 48px;
```

```
    min-width: 48px;
```

```
    padding: 12px;
```

```
}
```

```
/* Espaciado entre elementos táctiles */
```

```
.carousel-dot {
```

```
    margin: 8px;
```

```
    width: 12px;
```

```
    height: 12px;
```

```
}
```

```
```
```

### ### 4.2 Reducir Motion (Accesibilidad)

```
```css
```

```
@media (prefers-reduced-motion: reduce) {  
  
  *, *::before, *::after {  
  
    animation-duration: 0.01ms !important;  
    transition-duration: 0.01ms !important;  
  
  }  
  
}
```

```
```
```

### ### 4.3 Optimización para Conexiones Lentas (4G)

```
```javascript
```

```
// Detectar conexión lenta y servir imágenes LQIP
```

```
const connection = navigator.connection;
```

```
if (connection && (connection.saveData || connection.effectiveType.includes('2g'))) {
```

```
  document.documentElement.classList.add('data-saver');
```

```
}
```

```
// En CSS
```

```
.data-saver img {  
  filter: blur(5px);  
  transition: filter 0.3s;  
}  
  
.data-saver img.loaded {  
  filter: none;  
}  
  
...  
  
---
```

🚀 IMPLEMENTACIÓN RÁPIDA (Checklist)

Semana 1: Impacto Inmediato

- [] Reemplazar iframes de YouTube con facade/lite-youtube
- [] Optimizar logo (generar 200w, 400w, 800w)
- [] Implementar srcset en todas las imágenes
- [] Inline CSS crítico (< 14KB)

Semana 2: Optimización JS

- [] Implementar code splitting por rutas
- [] Lazy load componentes pesados (React.lazy)
- [] Eliminar console.logs y código muerto

- [] Agregar preconnect/dns-prefetch

Semana 3: Polish

- [] Service Worker para offline
- [] Fix accesibilidad (aria-labels, contraste)
- [] Optimizar fuentes (font-display: swap)
- [] Testing en dispositivos reales (iPhone, Android gama baja)

RESULTADOS ESPERADOS

| Métrica | Antes | Después | Mejora |

|-----|-----|-----|-----|

| ****Performance Score**** | 56 | 95-100 | +44 |

| ****LCP**** | 6.6s | 1.8s | -73% |

| ****TBT**** | 440ms | <150ms | -66% |

| ****Page Size**** | 3MB | <800KB | -73% |

| ****3G Load Time**** | ~12s | <3s | -75% |

HERRAMIENTAS RECOMENDADAS

1. **Imágenes:** Squoosh.app, Sharp (Node.js), cwebp
2. **Bundle Analysis:** `npx vite-bundle-visualizer` o webpack-bundle-analyzer
3. **Testing:** WebPageTest (dispositivos reales), Lighthouse CI
4. **Monitoreo:** Vercel Analytics, Cloudflare Web Analytics

📊 Informe Técnico de Optimización para grupotranservica.com

Objetivo: Alcanzar 100/100 en Lighthouse (Mobile + Desktop) con enfoque en velocidad móvil y compatibilidad multi-dispositivo

🟡 Estado Actual (9 feb 2026)

| Métrica | Móvil | Desktop | Objetivo |

|-----|-----|-----|-----|

| **Rendimiento** | 56/100 | 71/100 | **100/100** |

| FCP | 2.8s | 0.7s | <1.8s |

| LCP | 6.6s 🚩 | 2.1s | <2.5s |

| TBT | 440ms 🚩 | 400ms | <200ms |

| CLS | 0 | 0.018 | <0.1 |

| Tamaño total red | 3,031 KiB | 2,877 KiB | <1,500 KiB |

🚀 Plan de Acción Priorizado (Por Impacto)

✅ **NIVEL 1: CRÍTICO (Impacto >30 puntos)**

| Acción | Ahorro Estimado | Implementación |

|-----|-----|-----|

| **1. Optimizar imágenes hero** | 323 KiB + LCP -3.5s | • Convertir a WebP/AVIF con calidad 85%
• Generar `*srcset*` con tamaños: 320w, 640w, 1024w, 1920w
Ejemplo logo: Reducir de 1738x1482 → versión 200w para móvil + 400w desktop |

| **2. Diferir YouTube embeds** | 1,414 KiB + TBT -400ms | • Reemplazar iframes con `*preview thumbnails*`
• Cargar player solo al hacer clic (lazy-load)
• Usar librería lightweight: [lite-youtube-embed](https://github.com/paulirish/lite-youtube-embed) |

| **3. Eliminar JS/CSS no usado** | 1,020 KiB | • Code splitting por rutas (React.lazy + Suspense)
• Eliminar librerías no usadas en build
• Usar PurgeCSS para eliminar estilos no utilizados |

⚠️ **NIVEL 2: ALTO (Impacto 15-25 puntos)**

| Acción | Beneficio | Implementación |

|-----|-----|-----|

| **4. Eliminar bloqueo de render** | FCP -1.2s | • Inlinar CSS crítico (<10KB)
• Diferir CSS no crítico con ``media="print"` + onload`
• Preconectar a fonts.gstatic.com: `<link rel="preconnect" href="https://fonts.gstatic.com">`` |

| **5. Optimizar LCP** | LCP -1.8s | • Añadir ``fetchpriority="high"`` a imagen hero
• Precargar imagen LCP: `<link rel="preload" as="image" href="/hero.webp">``
• Eliminar

lazy-loading en imagen principal |

| ****6. Reducir TBT**** | TBT -250ms | • Partir bundle JS >50KB en chunks
• Mover YouTube analytics a web worker
• Usar `requestIdleCallback` para tareas no críticas |

🛠️ ****NIVEL 3: MEDIO (Impacto 5-15 puntos)****

| Acción | Beneficio | Implementación |

|-----|-----|-----|

| ****7. Configurar caché agresivo**** | Retención +20% | • Assets estáticos: `Cache-Control: max-age=31536000, immutable`
• HTML: `max-age=0, must-revalidate` |

| ****8. Corregir accesibilidad**** | WCAG AA | • Añadir `aria-label` a botones sin texto
• Aumentar contraste: verde #155d29 → #0d4a1e (ratio 4.5:1)
• Touch targets mínimos 48x48px |

| ****9. Seguridad CSP**** | Prevención XSS | • Reemplazar `unsafe-inline` por nonces:
`<script nonce="GENERADO_EN_RUNTIME">`
• Añadir `require-trusted-types-for 'script'` |

📱 Estrategia Multi-Dispositivo (Mobile-First)

```html

<!-- Patrón recomendado para imágenes responsivas -->

<picture>

<source media="(max-width: 768px)" srcset="/hero-768.webp" width="768" height="432">

<source media="(max-width: 1024px)" srcset="/hero-1024.webp" width="1024"

```
height="576">
```

```

```

```
</picture>
```

```
...
```

```
Viewport crítico:
```

```
```html
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, viewport-fit=cover">
```

```
...
```

```
---
```

```
### 📊 Roadmap de Implementación
```

```
| Semana | Acciones | Métrica Esperada |
```

```
|-----|-----|-----|
```

```
| **1** | Optimización imágenes + lazy-load YouTube | Móvil: 75+ / Desktop: 85+ |
```

| ****2**** | Code splitting + CSS crítico | Móvil: 85+ / Desktop: 92+ |

| ****3**** | Caché + accesibilidad + CSP | Móvil: 95+ / Desktop: 98+ |

| ****4**** | Ajustes finos + pruebas reales | ****100/100 ambos**** |

Configuración Técnica Recomendada

****Next.js/Nuxt (ejemplo):****

```
```.js
```

```
// next.config.js
```

```
module.exports = {
```

```
 images: {
```

```
 formats: ['image/avif', 'image/webp'],
```

```
 minimumCacheTTL: 31536000,
```

```
 deviceSizes: [320, 640, 768, 1024, 1280, 1920],
```

```
 },
```

```
 experimental: {
```

```
 optimizeCss: true,
```

```
 optimizeImages: true,
```

```
 }
```

```
}
```

```
...
```

**\*\*Headers críticos (nginx/apache):\*\***

```nginx

Assets estáticos

location ~* \.(js|css|webp|avif)\$ {

add_header Cache-Control "public, max-age=31536000, immutable";

}

HTML dinámico

location / {

add_header Cache-Control "public, max-age=0, must-revalidate";

}

...

 Checklist Final para 100/100

- [] Todas las imágenes <100KB en móvil (hero incluido)
- [] TBT < 200ms (medido en Moto G Power emulado)
- [] LCP < 2.5s en 3G lenta
- [] CLS = 0 (reservar espacio para iframes/embeds)
- [] JS total < 300KB en primera carga
- [] Validación WCAG 2.1 AA (axe-core)

- [] CSP con nonces + Trusted Types
- [] Pruebas en 5 dispositivos reales (iOS/Android)

> 💡 ****Consejo clave:**** Prioriza siempre el ***Core Web Vitals*** (LCP, FID/TBT, CLS). Un 90+ en estas 3 métricas garantiza >90 en puntuación general.

Informe de Optimización para el Sitio Web <https://grupotranservica.com>

Resumen Ejecutivo

Basado en los resultados de PageSpeed Insights proporcionados (capturados el 9 de febrero de 2026), el sitio web <https://grupotranservica.com> tiene un puntaje de rendimiento de ****56/100 en móvil**** y ****71/100 en escritorio****. Las métricas clave muestran oportunidades significativas de mejora:

- ****Móvil****: FCP (2.8 s, +6 pts), LCP (6.6 s, +2 pts), TBT (440 ms, +19 pts), CLS (0, +25 pts), SI (6.0 s, +5 pts). Puntaje total: 56.
- ****Escritorio****: FCP (0.7 s, +10 pts), LCP (2.1 s, +15 pts), TBT (400 ms, +13 pts), CLS (0.018, +25 pts), SI (1.5 s, +8 pts). Puntaje total: 71.

Otros puntajes: Accesibilidad (92 en móvil, 88 en escritorio), Prácticas Recomendadas (69 en ambos), SEO (100 en ambos).

El objetivo es elevar el puntaje de rendimiento a ****100/100**** en ambos entornos, priorizando la velocidad en dispositivos móviles (como se solicitó: apertura rápida y ajuste para todo tipo de móviles). Las principales causas son imágenes no optimizadas, JavaScript de terceros (especialmente YouTube), cadenas de solicitudes críticas y código no utilizado. Implementando las recomendaciones, se puede reducir el tiempo de carga inicial a <1 s en escritorio y <3 s en móvil, mejorando la experiencia en móviles con redes lentas (ej. 4G).

****Estimación de impacto****: Con optimizaciones, el puntaje podría subir a 90-100 en 1-2

semanas de trabajo, dependiendo del framework (parece ser un sitio basado en JavaScript, posiblemente React o similar con assets en /assets/).

Análisis de Métricas Clave

Usando los datos de Lighthouse 13.0.1:

| Métrica Móvil Escritorio Umbral Óptimo Impacto en Puntaje |
|---|
|---|

| |
|-------------------------------|
| ----- ----- ----- ----- ----- |
|-------------------------------|

| |
|--|
| FCP (First Contentful Paint) 2.8 s 0.7 s <1.8 s Alto en móvil; retrasos por CSS y JS bloqueantes. |
|--|

| |
|---|
| LCP (Largest Contentful Paint) 6.6 s 2.1 s <2.5 s Crítico; imágenes grandes y lazy-loading en LCP. |
|---|

| |
|---|
| TBT (Total Blocking Time) 440 ms 400 ms <200 ms Alto; JS de YouTube y código propio. |
|---|

| |
|--|
| CLS (Cumulative Layout Shift) 0 0.018 <0.1 Bueno, pero optimizable en escritorio. |
|--|

| |
|--|
| SI (Speed Index) 6.0 s 1.5 s <3.4 s Mejorable con optimización de recursos. |
|--|

Problemas comunes:

- **Imágenes:** Archivos grandes (ej. /assets/4_1754167186024-CI4L8huF.webp: 366 KiB en móvil). Ahorro estimado: 323 KiB en móvil, 310 KiB en escritorio.
- **Terceros:** YouTube consume ~1.7 MB y ~1 s de CPU, causando TBT alto.
- **JS/CSS no usado:** ~1 MB de JS no utilizado (principalmente YouTube), ~14 KiB de CSS.
- **Cadenas críticas:** Latencia máxima de 423 ms en móvil.
- **Accesibilidad:** Botones sin nombres accesibles, bajo contraste.
- **Prácticas:** Cookies de terceros, errores en consola (permisos), relación de aspecto incorrecta en imágenes.

- **SEO**: Estructura válida, pero checks manuales pendientes (datos estructurados).

Recomendaciones Detalladas

Priorizadas por impacto (alto a bajo). Enfocadas en móviles: usar responsive design, preload clave, y pruebas en emuladores (Moto G Power para móvil).

1. Optimizaciones de Rendimiento (Objetivo: 100/100)

- **Optimizar Imágenes** (Ahorro: 300-350 KiB, Impacto: LCP/FCP/SI +20-30 pts):
 - Comprimir y redimensionar imágenes con herramientas como ImageOptim o WebP (ya usan WebP, pero oversize). Ej: Reducir /assets/2_1754167....webp de 261 KiB a <50 KiB para vistas móviles (usa srcset para tamaños adaptables: mobile=768px, desktop=1920px).
 - Evitar lazy-loading en LCP (logo: /assets/logo%20transervica%20sin%20fondo_1754163034585-PZeMbfCv.webp). Agregar `loading="eager"` y `fetchpriority="high"`.
 - Implementar CDN (ej. Cloudflare) para servir imágenes optimizadas automáticamente.
 - Para móviles: Usar formatos AVIF si es compatible, y dimensiones responsive (evitar escalado innecesario; corrige relación de aspecto 1.17 vs. 1.33 en logo).
- **Reducir JS/CSS Bloqueante** (Impacto: TBT/FCP +15-20 pts):
 - Mover CSS crítico inline (<15 KiB de /assets/index-D1hjCskL.css). Usar critical CSS tools como Penthouse.
 - Defer o async JS no esencial (ej. /assets/index-BiVAZ_yN.js: 121 KiB). Eliminar JS no usado (~37 KiB ahorro).
 - Para YouTube: Reemplazar embeds con lite-embeds (ej. lite-youtube-embed library) o lazy-load iframes. Cargar solo al interactuar (reduce ~1 MB y 1 s de CPU).
 - Minificar y tree-shake JS/CSS (usa herramientas como Terser o PurgeCSS).

- ****Cadenas Críticas y Preconexiones (Impacto: LCP/SI +10 pts)**:**
 - Preconnect a terceros: Agregar `<link rel="preconnect" href="https://www.youtube.com">` y `<link rel="preconnect" href="https://fonts.gstatic.com">`.
 - Preload LCP: `<link rel="preload" as="image" href="/assets/logo....webp">`.
 - Reducir cadenas: Posponer JS no crítico (ej. `/assets/button-Jf03mtiP.js`).
- ****Caché y Entrega (Impacto: General +5-10 pts)**:**
 - Aumentar TTL de caché: Para ads (DoubleClick: 15 min → 1 día). Usa service workers para caché offline en móviles.
 - Servir desde HTTP/2 o HTTP/3 para multiplexado.
 - Para móviles: Habilitar AMP si aplica, o usar PWA para carga rápida en redes lentas.
- ****Fuentes y CLS (Impacto: CLS/FCP +5 pts)**:**
 - Usar `font-display: swap` en Google Fonts para evitar FOIT.
 - Preload fuentes: `<link rel="preload" as="font" href=".../KFO7CnqEu....woff2" crossorigin>`.
- ****Tareas Largas y TBT (Impacto: TBT +15 pts)**:**
 - Dividir JS en chunks (usa Webpack code-splitting).
 - Optimizar YouTube: Limitar a 1 embed visible; usar API para pausar.
- ****Cargas Útiles Grandes (Impacto: General)**:**
 - Reducir total <1.5 MB (actual: 3 MB en móvil). Comprimir con Brotli/Gzip.

2. ****Mejoras en Accesibilidad (Objetivo: 100/100)****

- ****Nombres Accesibles****: Agregar `aria-label` a botones (ej. carrusel: `- ****Contraste****: Aumentar ratio a >4.5:1 (ej. texto blanco en verde #3ea30f → usar #fff con fondo más oscuro o viceversa). Usa herramientas como WAVE.
- ****Áreas Táctiles (Móvil)****: Aumentar botones a >48x48 px con padding (ej. dots en carrusel).
- ****Manuales****: Verificar ARIA roles, foco teclado, y alt texts descriptivos. Pruebas con screen readers (VoiceOver/Android TalkBack).

3. ****Prácticas Recomendadas (Objetivo: 100/100)****

- ****Cookies de Terceros****: Migrar a first-party o usar Partitioned cookies (CHIPS) para YouTube.
- ****Errores en Consola****: Corregir violaciones de permissions policy (agregar `allow="accelerometer; gyroscope"` en iframes si needed).
- ****Issues Panel****: Resolver cookies en embeds YouTube (usa `sameSite=None; Secure`).
- ****Relación de Aspecto****: Especificar width/height en imgs para mantener ratio (ej. `aspect-ratio: 1.17` en CSS).
- ****CSP Efectiva****: Agregar CSP header: `Content-Security-Policy: script-src 'self' 'nonce-...' https://www.youtube.com;`. Usar Trusted Types para DOM XSS.

4. ****SEO (Mantener 100/100)****

- ****Manuales****: Validar datos estructurados con Google's Structured Data Testing Tool. Agregar schema para empresa/transporte.
- ****Móvil****: Asegurar viewport: `

Plan de Acción Priorizado

1. **Inmediato (1-3 días, +20-30 pts)**: Optimizar imágenes y LCP (comprimir, eager load). Pruebas en móvil.
2. **Corto Plazo (3-7 días, +20 pts)**: Reducir JS de YouTube, inline CSS crítico.
3. **Medio Plazo (1-2 semanas, +10-20 pts)**: CSP, accesibilidad, caché. Re-ejecutar PageSpeed.
4. **Pruebas**: Usa Lighthouse en Chrome DevTools. Prueba en dispositivos reales (Android/iOS, redes 4G lenta). Monitorea con Google Search Console.

Beneficios Esperados

- **Velocidad Móvil**: FCP <2 s, LCP <3 s, apertura en <5 s en 4G.
- **Ajuste Móvil**: Responsive con media queries; prueba en rangos (320px-1024px).
- **Puntaje Final**: 95-100/100 en rendimiento. Mejora SEO indirecta (baja tasa de rebote), accesibilidad para más usuarios.
- **Herramientas**: GTmetrix, WebPageTest para validación. Si es CMS (ej. WordPress), usa plugins como Autoptimize.

Informe Exhaustivo de Optimización para el Sitio Web <https://grupotranservica.com>

Introducción y Contexto

Este informe se basa en los análisis de PageSpeed Insights realizados el 9 de febrero de 2026 (capturados a las 17:53 GMT-5), utilizando Lighthouse 13.0.1 en entornos emulados: Moto G Power para móvil (con limitación de 4G lenta) y escritorio (con limitación personalizada). El sitio web, aparentemente construido con un framework JavaScript moderno (posiblemente Vite o React, dado los assets como /assets/index-BiVAZ_yN.js y CSS empaquetado), presenta oportunidades significativas de mejora para alcanzar un puntaje de rendimiento del 100/100 en ambos dispositivos.

Puntajes Actuales:

- ****Móvil****: Rendimiento 56/100, Accesibilidad 92/100, Prácticas Recomendadas 69/100, SEO 100/100.

- ****Escritorio****: Rendimiento 71/100, Accesibilidad 88/100, Prácticas Recomendadas 69/100, SEO 100/100.

El enfoque principal, como solicitado, es optimizar para dispositivos móviles: carga rápida en redes lentas (ej. 4G), compatibilidad con una amplia gama de dispositivos (desde iPhone SE hasta Galaxy S-series, con resoluciones de 320px a 1024px), y ajuste responsive para pantallas táctiles. Esto incluye minimizar el uso de datos, reducir latencia y asegurar accesibilidad en móviles con lectores de pantalla como TalkBack o VoiceOver.

****Objetivos****:

- Elevar rendimiento a 100/100: Reducir FCP <1.8s, LCP <2.5s, TBT <200ms, CLS <0.1, SI <3.4s.

- Mejorar accesibilidad a 100/100: Corregir nombres, contraste y áreas táctiles.

- Elevar prácticas recomendadas a 100/100: Resolver cookies, errores y CSP.

- Mantener SEO en 100/100 con mejoras manuales.

- ****Impacto Estimado****: Reducción de tamaño de página de ~3MB a <1.5MB, tiempo de carga inicial <3s en móvil 4G. Mejora en tasas de conversión (ej. solicitudes de cotización) hasta 20-30% por mejor UX.

****Metodología****: Análisis de métricas, auditorías y diagnósticos de Lighthouse. Recomendaciones incluyen pasos accionables, ejemplos de código y herramientas. Asumo acceso al código fuente; si no, usa un CMS como WordPress con plugins equivalentes.

Análisis Detallado de Métricas

Las métricas se calculan directamente de los datos de carga inicial. Usan emulación, pero varían en dispositivos reales (prueba con WebPageTest.org para validación).

| Métrica | Descripción | Móvil (Valor/Puntos) | Escritorio (Valor/Puntos) | Umbral Óptimo (Core Web Vitals) | Causas Principales Identificadas |

|-----|-----|-----|-----|-----|
-----|-----|

| **FCP (First Contentful Paint)** | Tiempo hasta el primer renderizado de contenido (texto/imagen). | 2.8s (+6 pts) | 0.7s (+10 pts) | <1.8s (bueno), <1s (ideal) | CSS bloqueante (/assets/index-D1hjCskL.css: 19KiB, 310ms), JS de terceros (YouTube), fuentes sin swap. En móvil, agravado por 4G lenta. |

| **LCP (Largest Contentful Paint)** | Tiempo hasta renderizado del elemento más grande (ej. logo o imagen hero). | 6.6s (+2 pts) | 2.1s (+15 pts) | <2.5s (bueno), <1.5s (ideal) | Imágenes oversize (ej. logo: 105KiB para 197x168px), lazy-loading en LCP, cadenas críticas (423ms en móvil). Elemento LCP: . Desglose: TTFB 0ms, carga recurso 350ms (móvil). |

| **TBT (Total Blocking Time)** | Tiempo que el hilo principal está bloqueado por JS (>50ms por tarea). | 440ms (+19 pts) | 400ms (+13 pts) | <200ms (bueno), <100ms (ideal) | JS de YouTube (1.5s CPU, ~1MB no usado), código propio (/assets/index-BiVAZ_yN.js: 300ms eval). 7 tareas largas (>50ms), ej. base.js de YouTube: 196ms. |

| **CLS (Cumulative Layout Shift)** | Cambios inesperados en layout (ej. por fuentes o ads). | 0 (+25 pts) | 0.018 (+25 pts) | <0.1 (bueno), 0 (ideal) | Bueno en móvil; en escritorio, causado por fuente web (/UcC73FwrK....woff2: 10ms ahorro con swap). Elementos: <div class="flex items-center justify-between py-3">. |

| **SI (Speed Index)** | Velocidad percibida de carga visual. | 6.0s (+5 pts) | 1.5s (+8 pts) | <3.4s (bueno), <2s (ideal) | Combinado de arriba; imágenes y JS retrasan renderizado. |

Diagnósticos Adicionales:

- **Tiempo de Ejecución JS:** 1.6s móvil, 1.4s escritorio. Mayor en YouTube (731ms eval en móvil).
- **Trabajo Hilo Principal:** 2.8s móvil (Script Eval 1488ms), 2.7s escritorio.
- **JS/CSS No Usado:** ~1006KiB JS (968KiB de YouTube), ~14KiB CSS.
- **Cargas Grandes:** Total 3031KiB móvil, 2877KiB escritorio. Principales: YouTube base.js (421KiB x2).

- **Terceros**: YouTube (1718KiB, 936ms hilo en móvil; 1694KiB, 1060ms en escritorio).
- **User Timings**: 68 marcas en móvil (ej. mark_fs: 1078ms), 52 en escritorio. Indican retrasos en init (ep_init_pr: 1153ms móvil).

Recomendaciones Detalladas por Categoría

1. Optimizaciones de Rendimiento (Prioridad Alta: Impacto +40-50 pts)

Objetivo: Reducir latencia en móviles con 4G (TTFB <200ms, descarga <1MB).

1.1 Optimización de Imágenes (Ahorro: 323KiB móvil, 310KiB escritorio; Impacto: LCP/FCP/SI)

- **Problemas Específicos**: Archivos oversize para dimensiones mostradas (ej. /assets/2_1754167....webp: 261KiB para 665x499px móvil, debería <100KiB). Relación de aspecto incorrecta en logo (1.33 vs. 1.17 real), causando reflows. No adaptables a viewport móvil.

- **Pasos Accionables**:

1. **Compresión y Formatos**: Usa herramientas como TinyPNG o ImageMagick para comprimir WebP a 80% calidad. Para AVIF (mejor compresión): Convierte con `convert input.webp -quality 75 output.avif`. Prueba compatibilidad (Safari 16+).

2. **Imágenes Responsivas**: Implementa srcset y sizes en :

```

<<html


...

```

- Genera variantes: Small (320x240, <20KiB para móviles low-end), Medium (768x576,

<50KiB), Large (1920x1080, <200KiB).

3. ****Evitar Lazy en LCP****: Cambia loading="lazy" a "eager" para elementos above-the-fold (logo, hero images). Usa IntersectionObserver para lazy manual en JS si needed.

4. ****Corrección de Aspecto****: Agrega CSS `aspect-ratio: 1738 / 1482;` o atributos width/height reales para prevenir CLS.

5. ****CDN y Automatización****: Integra Cloudinary o Imgix para optimización automática: `https://res.cloudinary.com/demo/image/fetch/f_auto,q_auto,w_768/https://grupotranservica.com/assets/2_1754167....webp`.

6. ****Pruebas Móviles****: Usa Chrome DevTools (emular Moto G) para verificar LCP. Esperado: Reducción LCP a <2s en móvil.

- ****Impacto en Móviles****: Reduce datos en 4G (ahorro 300KiB → <1s descarga). Ajuste para dispositivos: Usa media queries para servir smaller en portrait mode.

1.2 Reducción de JS/CSS Bloqueante (Impacto: TBT/FCP +20 pts)

- ****Problemas****: /assets/index-D1hjCskL.css bloquea 310ms; JS no usado (37KiB en propio, 968KiB en YouTube).

- ****Pasos****:

1. ****CSS Crítico Inline****: Extrae CSS above-the-fold con Critical (npm tool): Inline <style> en <head> (~5KiB). Defer resto: `<link rel="stylesheet" href="..." media="print" onload="this.media='all'">`.

2. ****JS Optimización****: Minifica con Terser. Code-split: En Vite/Webpack, usa dynamic imports: `const About = () => import('./about-section.js');`. Elimina no usado con Tree Shaking.

3. ****YouTube Lite****: Reemplaza iframes con <lite-youtube> (web component):

```
```html
```

```
<lite-youtube videoid="_LQbWkWlg6s" playlabel="Play: Proyecto Video"></lite-youtube>
```

```
...
```

- Carga solo thumbnail inicialmente (ahorro 1.7MB). Usa IntersectionObserver para load on view.

4. **\*\*Defer No Esencial\*\***: Agrega defer/async: ``<script src="/assets/index-BiVAZ_yN.js" defer></script>``.

5. **\*\*Pruebas\*\***: Monitorea TBT con Performance panel en DevTools. Esperado: TBT <100ms.

#### #### 1.3 Cadenas Críticas y Preconexiones (Impacto: LCP/SI +15 pts)

- **\*\*Problemas\*\***: Latencia 423ms móvil (cadenas como `/index.js` → `/projects.js`).

- **\*\*Pasos\*\***:

1. **\*\*Preconnect/Preload\*\***: En `<head>`:

```
```html
```

```
<link rel="preconnect" href="https://www.youtube.com" crossorigin>
```

```
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

```
<link rel="preload" as="image" href="/assets/logo....webp" fetchpriority="high">
```

```
<link rel="preload" as="font" href=".../KFO7CnqEu....woff2" type="font/woff2" crossorigin>
```

```
...
```

2. ****Reducir Cadenas****: Posponer JS innecesario (ej. `/assets/faq-section-BuRQDHar.js` al final).

3. ****HTTP/3****: Configura server (Nginx/Apache) para HTTP/3. Usa Brotli para compresión (mejor que Gzip).

1.4 Caché y Entrega (Impacto: General +10 pts)

- ****Problemas****: TTL corto (15min para ads).

- ****Pasos****: En `.htaccess` o server config: ``ExpiresByType image/webp "access plus 1 year"``.
Implementa Service Worker:

```
```js
```

```
// sw.js

self.addEventListener('fetch', event => {

 event.respondWith(caches.match(event.request).then(response => response ||
fetch(event.request)));

});

...

```

- Registra en main.js: `navigator.serviceWorker.register('/sw.js');`.
- **Móviles**: PWA manifest para instalación, caché offline.

#### #### 1.5 Fuentes y CLS (Impacto: +5 pts)

- **Pasos**: Agrega a `@font-face: `font-display: swap;``. Preload como arriba. Usa system fonts fallback: `font-family: Inter, system-ui;``.

#### #### 1.6 Tareas Largas y Diagnósticos Adicionales

- **Pasos**: Divide tareas largas (ej. YouTube base.js) con Web Workers. Monitorea con Sentry o New Relic.

### ### 2. Mejoras en Accesibilidad (Prioridad Media: Impacto a 100/100)

- **Botones Sin Nombres**: Agrega `aria-label` a todos `<button>` (ej. carrusel: `aria-label="Ir a proyecto 1"`). Para dots: Usa `<ul role="tablist">` con `<li role="tab">`.
- **Contraste Bajo**: En botones verdes (#155d29, texto #fff): Ratio 4.5:1 mínimo. Cambia a #0d3a1a o usa herramientas como Color Contrast Analyzer.
- **Áreas Táctiles (Móvil)**: CSS: `.button { min-width: 48px; min-height: 48px; padding: 12px; }`. Prueba con Lighthouse Accessibility.
- **Manuales**: Verifica con WAVE tool: ARIA landmarks, keyboard nav (no traps), alt texts (descriptivos, ej. "Logo de TRANSERVICA transportando cargas excepcionales en Venezuela").

- **Móviles**: Asegura zoom (viewport meta), touch targets. Pruebas: Emula en iOS/Android.

### ### 3. Prácticas Recomendadas (Prioridad Media: Impacto a 100/100)

- **Cookies Terceros**: Migra YouTube a first-party o usa `Partitioned` en Set-Cookie.
- **Errores Consola**: Agrega a iframes: `allow="accelerometer; gyroscope"`. Corrige permissions policy: Header `Permissions-Policy: accelerometer=(), gyroscope=()`. Prueba con CSP Evaluator.
- **Issues Panel**: Para cookies: Agrega `SameSite=None; Secure` en embeds.
- **Relación Aspecto**: Como en 1.1.
- **CSP**: Header: `Content-Security-Policy: default-src 'self'; script-src 'self' https://www.youtube.com 'nonce-abc'; require-trusted-types-for 'script';`. Prueba con CSP Evaluator.
- **Trusted Types**: Implementa en JS: `const policy = trustedTypes.createPolicy('default', { createHTML: s => s });`.

### ### 4. SEO (Mantener/MEjorar)

- **Manuales**: Valida schema.org con Structured Data Testing Tool: Agrega JSON-LD para Organization y Service.
- **Móviles**: Asegura mobile-first design. Prueba Google Mobile-Friendly: Evita popups intrusivos.

## ## Plan de Implementación y Pruebas

- **Fase 1 (1-3 días)**: Imágenes y LCP (pruebas locales con npm run build).
- **Fase 2 (4-7 días)**: JS/CSS, YouTube.
- **Fase 3 (8-14 días)**: Accesibilidad, CSP, caché.
- **Herramientas**: Lighthouse CLI (`lighthouse https://grupotranservica.com --output html`), GTmetrix, WebPageTest (emular móviles específicos).

- **\*\*Monitoreo\*\***: Google Analytics para bounce rate; Search Console para Core Web Vitals.
- **\*\*Pruebas Móviles\*\***: Dispositivos reales (Android 10+, iOS 15+), redes simuladas (DevTools throttling). Asegura apertura en <3s en 4G, ajuste a rotación (CSS @media orientation).

# 📄 PLAN DETALLADO DE OPTIMIZACIÓN - grupotranservica.com

## 🎯 Objetivo Final: 100/100 Lighthouse (Mobile + Desktop)

---

## 📱 FASE 1: OPTIMIZACIÓN DE IMÁGENES (Semana 1)

### 1.1 Optimizar Imagen Hero Principal

**\*\*Problema:\*\*** `/assets/4\_1754167186024-CI4L8huF.webp` - 366.5 KiB, mostrada a 665x499px

```bash

Instalar herramientas

npm install -g sharp-cli imagemin-cli

Convertir y redimensionar

sharp convert "4_1754167186024-CI4L8huF.webp" \

--resize 1920 1080 --quality 85 --format webp \

--output "hero-1920.webp"

```
sharp convert "4_1754167186024-CI4L8huF.webp" \  
  --resize 1024 576 --quality 85 --format webp \  
  --output "hero-1024.webp"
```

```
sharp convert "4_1754167186024-CI4L8huF.webp" \  
  --resize 768 432 --quality 80 --format webp \  
  --output "hero-768.webp"
```

```
sharp convert "4_1754167186024-CI4L8huF.webp" \  
  --resize 640 360 --quality 80 --format webp \  
  --output "hero-640.webp"
```

```
sharp convert "4_1754167186024-CI4L8huF.webp" \  
  --resize 320 180 --quality 75 --format webp \  
  --output "hero-320.webp"
```

...

****Implementación HTML:****

```
```html
```

```
<picture>
```

```
 <source media="(min-width: 1280px)" srcset="/assets/hero-1920.webp" width="1920"
 height="1080">
```

```
 <source media="(min-width: 1024px)" srcset="/assets/hero-1024.webp" width="1024"
 height="576">
```

```
<source media="(min-width: 768px)" srcset="/assets/hero-768.webp" width="768" height="432">
```

```
<source media="(min-width: 640px)" srcset="/assets/hero-640.webp" width="640" height="360">
```

```

```

```
</picture>
```

```
...
```

### ### 1.2 Optimizar Logo

**\*\*Problema:\*\*** Logo de 1738x1482px (105.2 KiB) mostrado a 197x168px

```
```bash
```

```
# Crear versiones optimizadas
```

```
sharp convert "logo transervica sin fondo.webp" \
```

```
--resize 400 342 --quality 90 --format webp \
```

```
--output "logo-400.webp"
```

```
sharp convert "logo transervica sin fondo.webp" \  
  --resize 200 171 --quality 85 --format webp \  
  --output "logo-200.webp"
```

```
sharp convert "logo transervica sin fondo.webp" \  
  --resize 100 86 --quality 80 --format webp \  
  --output "logo-100.webp"
```

```
...
```

****Implementación HTML:****

```
```html
```

```
<picture>
```

```
 <source media="(min-width: 1024px)" srcset="/assets/logo-400.webp" width="400"
 height="342">
```

```
 <source media="(min-width: 640px)" srcset="/assets/logo-200.webp" width="200"
 height="171">
```

```

```

```
</picture>
```

```
...
```



### ### 1.3 Optimizar Todas las Imágenes del Proyecto

```
```javascript
```

```
// Script de optimización masiva
```

```
const sharp = require('sharp');
```

```
const fs = require('fs');
```

```
const path = require('path');
```

```
const optimizeImage = async (inputPath, outputPath, maxWidth, quality = 85) => {
```

```
  const metadata = await sharp(inputPath).metadata();
```

```
  const width = Math.min(metadata.width, maxWidth);
```

```
  const height = Math.round(width * metadata.height / metadata.width);
```

```
  await sharp(inputPath)
```

```
    .resize(width, height, { fit: 'inside', withoutEnlargement: true })
```

```
    .webp({ quality: quality })
```

```
    .toFile(outputPath);
```

```
  console.log(`✓ ${path.basename(inputPath)}: ${width}x${height} @ ${quality}%`);
```

```
};
```

```
// Optimizar todas las imágenes
```

```
const imagesDir = './public/assets';
```

```

const optimizedDir = './public/assets/optimized';

fs.readdirSync(imagesDir).forEach(async (file) => {
  if (file.match(/\.(webp|jpg|jpeg|png)$/i)) {
    const inputPath = path.join(imagesDir, file);
    const outputPath = path.join(optimizedDir, file);

    await optimizeImage(inputPath, outputPath, 1920, 85);
  }
});
...

---
```

🧑‍💻 FASE 2: OPTIMIZACIÓN YOUTUBE EMBEDS (Semana 1-2)

2.1 Reemplazar Iframes con Lazy-Load

****Crear componente LiteYouTubeEmbed:****

```

```javascript
// components/LiteYouTubeEmbed.jsx

import { useState } from 'react';
```

```
const LiteYouTubeEmbed = ({ videoId, title, posterUrl, autoPlay = false }) => {

 const [loaded, setLoaded] = useState(false);

 const handleClick = () => {

 setLoaded(true);

 };

 return (

 <div className="lite-youtube relative w-full max-w-[1280px] mx-auto">

 {!loaded ? (

 <button

 onClick={handleClick}

 className="w-full h-full relative bg-black cursor-pointer"

 aria-label={`Ver video: ${title}`}

 >

 {/* Imagen de previsualización */}

 <img

 src={posterUrl || `https://i.ytimg.com/vi/${videoId}/hqdefault.jpg`}

 alt={title}

 className="w-full h-auto"

 loading="lazy"

 />

)
```

```

 { /* Botón de play */ }

 <div className="absolute inset-0 flex items-center justify-center">

 <div className="w-20 h-20 bg-red-600 rounded-full flex items-center justify-center
opacity-90 hover:opacity-100 transition-opacity">

 <svg className="w-10 h-10 text-white" fill="currentColor" viewBox="0 0 24 24">

 <path d="M19.615 3.184c-3.604-.246-11.631-.245-15.23 0-3.897.266-4.356 2.62-4.385
8.816.029 6.185.484 8.549 4.385 8.816 3.6.245 11.626.246 15.23 0 3.897-.266 4.356-2.62
4.385-8.816-.029-6.185-.484-8.549-4.385-8.816zm-10.615 12.816v-8l8 3.993-8 4.007z"/>

 </svg>

 </div>

 </div>

 { /* Texto "Play" */ }

 <div className="absolute bottom-2 left-2 bg-black bg-opacity-75 text-white text-sm
px-2 py-1 rounded">

 {title}

 </div>

</button>

): (

<iframe

 src={`https://www.youtube.com/embed/${videoid}?autoplay=${autoPlay ? 1 : 0}&rel=0`}

 title={title}

 allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope;
picture-in-picture"

 allowFullScreen

```

```

 className="w-full h-auto aspect-video"

 ></iframe>

)}

</div>

);

};

export default LiteYouTubeEmbed;
...

Uso en componentes:

```jsx
// En lugar de iframe directo
<LiteYouTubeEmbed
  videoId="_LQbWkWlg6s"
  title="Proyecto TRANSERVICA: Multicar Railway Convoy"
  posterUrl="/assets/thumbnails/railway-preview.jpg"
/>

<LiteYouTubeEmbed
  videoId="17qCAJDIZuM"
  title="Transporte de Cargas Excepcionales"
  posterUrl="/assets/thumbnails/cargas-preview.jpg"

```

```
/>
```

```
...
```

2.2 Precargar Thumbnails

```
```html
```

```
<!-- En el <head> -->
```

```
<link rel="preconnect" href="https://i.ytimg.com">
```

```
<link rel="preload" as="image" href="https://i.ytimg.com/vi/_LQbWkWlg6s/hqdefault.jpg">
```

```
<link rel="preload" as="image" href="https://i.ytimg.com/vi/17qCAJDIZuM/hqdefault.jpg">
```

```
...
```

```

```

## ## ⚡ FASE 3: OPTIMIZACIÓN DE JAVASCRIPT (Semana 2)

### ### 3.1 Code Splitting por Rutas

```
```javascript
```

```
// routes/_layout.jsx
```

```
import { Suspense, lazy } from 'react';
```

```
import { Outlet } from 'react-router-dom';
```

```
const ProjectsSection = lazy(() => import('../components/ProjectsSection'));
```

```
const ServicesSection = lazy(() => import('../components/ServicesSection'));
const AboutSection = lazy(() => import('../components/AboutSection'));
const EquipmentSection = lazy(() => import('../components/EquipmentSection'));
const BlogSection = lazy(() => import('../components/BlogSection'));
const ContactSection = lazy(() => import('../components/ContactSection'));
```

```
export default function Layout() {
  return (
    <div>
      <Header />
      <Suspense fallback={<LoadingSpinner />}>
        <Outlet />
      </Suspense>
      <Footer />
    </div>
  );
}
```

```
function LoadingSpinner() {
  return (
    <div className="flex items-center justify-center min-h-[400px]">
      <div className="animate-spin rounded-full h-16 w-16 border-b-4
border-green-600"></div>
    </div>
  );
}
```

```
</div>

);

}

...

```

3.2 Eliminar Código No Usado

```
````javascript

// vite.config.js o webpack.config.js

export default {

 build: {

 rollupOptions: {

 output: {

 manualChunks: {

 'react-vendor': ['react', 'react-dom', 'react-router-dom'],

 'ui-components': ['@headlessui/react', '@heroicons/react'],

 'utils': ['lodash', 'date-fns'],

 },

 },

 },

 chunkSizeWarningLimit: 500,

 minify: 'terser',

 terserOptions: {

```



```
compress: {
 drop_console: true,
 drop_debugger: true,
},

},

},
};
...
```

### ### 3.3 Diferir Scripts No Críticos

```
```html  
  
<!-- En el <head> -->  
  
<script type="module" src="/assets/index-BiVAZ_yN.js" defer></script>  
  
  
  
<!-- Scripts no críticos al final del body -->  
  
<script type="module" src="/assets/projects-section.js" async></script>  
  
<script type="module" src="/assets/services-section.js" async></script>  
  
<script type="module" src="/assets/equipment-section.js" async></script>  
  
...  
  
---
```

🎨 FASE 4: OPTIMIZACIÓN CSS (Semana 2)

4.1 Extraer CSS Crítico

```
```:javascript
// scripts/extract-critical-css.js

const fs = require('fs');
const path = require('path');
const { JSDOM } = require('jsdom');
const PurgeCSS = require('purgecss').PurgeCSS;

async function extractCriticalCSS() {
 const html = fs.readFileSync('./dist/index.html', 'utf8');
 const dom = new JSDOM(html);

 // Simular viewport móvil
 dom.window.document.documentElement.style.width = '375px';
 dom.window.document.documentElement.style.height = '667px';

 const criticalSelectors = [];

 // Extraer estilos visibles en el viewport inicial
 const visibleElements = dom.window.document.querySelectorAll(`
```

```
header, nav, .hero, .hero *,
[fetchpriority="high"],
img[loading="eager"],
button, a
`);
```

```
visibleElements.forEach(el => {
 const computed = dom.window.getComputedStyle(el);
 criticalSelectors.push(el.tagName.toLowerCase());
 if (el.className) {
 criticalSelectors.push(...el.className.split(' '));
 }
});
```

```
console.log('Critical selectors:', criticalSelectors);
```

```
// Usar PurgeCSS para extraer solo lo necesario
```

```
const purgeCSSResult = await new PurgeCSS().purge({
 content: ['./dist/**/*.html', './src/**/*.jsx'],
 css: ['./src/**/*.css'],
 safelist: criticalSelectors,
});
```

```
fs.writeFileSync(
 './dist/critical.css',
 purgeCSSResult[0].css
);

console.log('✓ Critical CSS extracted');
}

extractCriticalCSS();
...
```

### ### 4.2 Inlinar CSS Crítico + Diferir Resto

```
```html

<!-- En el <head> -->

<style>

/* CSS crítico inlinado (máximo 10KB) */

:root{--transervica-green:#155d29}

body{margin:0;font-family:Inter,sans-serif;color:#1f2937}

header{position:sticky;top:0;z-index:50;background:white;box-shadow:0 2px 4px
rgba(0,0,0,.1)}

nav{display:flex;justify-content:space-between;align-items:center;padding:1rem}

.logo{height:2rem}
```

```
.hero{position:relative;min-height:100vh;display:flex;align-items:center;justify-content:center}
```

```
.button{display:inline-block;background:var(--transervica-green);color:white;padding:0.75rem 1.5rem;border-radius:0.375rem;font-weight:600;text-decoration:none;transition:all 0.3s}
```

```
.button:hover{transform:scale(1.05);box-shadow:0 4px 6px rgba(21,93,41,.3)}
```

```
</style>
```

```
<!-- CSS no crítico diferido -->
```

```
<link rel="stylesheet" href="/assets/index-D1hjCskL.css" media="print"
onload="this.media='all'">
```

```
<noscript><link rel="stylesheet" href="/assets/index-D1hjCskL.css"></noscript>
```

```
...
```

```
---
```

🛠 FASE 5: CONFIGURACIÓN DE CACHE (Semana 3)

5.1 Headers HTTP para Caché

```
**Nginx:**
```

```
```nginx
```

```
server {
```

```
 listen 80;
```

```
 server_name grupotranservica.com;
```

# Assets estáticos con caché largo

```
location ~* \.(webp|avif|jpg|jpeg|png|gif|svg|ico)$ {
 expires 1y;
 add_header Cache-Control "public, immutable";
 access_log off;
}
```

```
location ~* \.(js|css|woff2|woff|ttf|otf)$ {
 expires 1y;
 add_header Cache-Control "public, immutable";
 access_log off;
}
```

# HTML con caché corto

```
location ~* \.(html|htm)$ {
 expires 0;
 add_header Cache-Control "public, must-revalidate";
}
```

# API y datos dinámicos

```
location /api/ {
 expires 0;
 add_header Cache-Control "no-cache, no-store, must-revalidate";
```

```
}
}
...
```

**\*\*Apache (.htaccess):\*\***

```
```apache
```

```
<IfModule mod_expires.c>
```

```
    ExpiresActive On
```

```
    # Imágenes - 1 año
```

```
    ExpiresByType image/webp "access plus 1 year"
```

```
    ExpiresByType image/avif "access plus 1 year"
```

```
    ExpiresByType image/jpeg "access plus 1 year"
```

```
    ExpiresByType image/png "access plus 1 year"
```

```
    ExpiresByType image/svg+xml "access plus 1 year"
```

```
    # CSS/JS - 1 año
```

```
    ExpiresByType text/css "access plus 1 year"
```

```
    ExpiresByType application/javascript "access plus 1 year"
```

```
    ExpiresByType application/x-javascript "access plus 1 year"
```

```
    # Fuentes - 1 año
```

```
    ExpiresByType font/woff2 "access plus 1 year"
```

```
ExpiresByType font/woff "access plus 1 year"
```

```
# HTML - sin caché
```

```
ExpiresByType text/html "access plus 0 seconds"
```

```
</IfModule>
```

```
<IfModule mod_headers.c>
```

```
# Assets inmutables
```

```
<FilesMatch "\.(webp|avif|jpg|jpeg|png|gif|svg|js|css|woff2|woff)$">
```

```
    Header set Cache-Control "public, max-age=31536000, immutable"
```

```
</FilesMatch>
```

```
# HTML revalidable
```

```
<FilesMatch "\.(html|htm)$">
```

```
    Header set Cache-Control "public, max-age=0, must-revalidate"
```

```
</FilesMatch>
```

```
</IfModule>
```

```
...
```

5.2 Service Worker para Caché Avanzado

```
````javascript
```

```
// public/sw.js
```



```
const CACHE_NAME = 'transervica-v1';
```

```
const urlsToCache = [
```

```
 '/',
```

```
 '/index.html',
```

```
 '/assets/critical.css',
```

```
 '/assets/logo-200.webp',
```

```
 '/assets/hero-640.webp',
```

```
];
```

```
self.addEventListener('install', (event) => {
```

```
 event.waitUntil(
```

```
 caches.open(CACHE_NAME)
```

```
 .then((cache) => {
```

```
 console.log('Opened cache');
```

```
 return cache.addAll(urlsToCache);
```

```
 })
```

```
);
```

```
 self.skipWaiting();
```

```
});
```

```
self.addEventListener('fetch', (event) => {
```

```
 event.respondWith(
```

```
 caches.match(event.request)
```

```
.then((response) => {
 if (response) {
 return response;
 }

 return fetch(event.request).then(
 (response) => {
 if (!response || response.status !== 200 || response.type !== 'basic') {
 return response;
 }

 const responseToCache = response.clone();

 caches.open(CACHE_NAME)
 .then((cache) => {
 cache.put(event.request, responseToCache);
 });

 return response;
 }
);
})
);
```

```

});

self.addEventListener('activate', (event) => {

 const cacheWhitelist = [CACHE_NAME];

 event.waitUntil(

 caches.keys().then((cacheNames) => {

 return Promise.all(

 cacheNames.map((cacheName) => {

 if (cacheWhitelist.indexOf(cacheName) === -1) {

 return caches.delete(cacheName);

 }

 })

);

 })

);

 self.clients.claim();

});

...

```

### ### 6.1 Botones con ARIA Labels

```
```jsx
```

```
// Botones del slider de proyectos
```

```
<button
```

```
  className="w-12 h-12 rounded-full bg-white border border-gray-200  
  hover:border-[#155d29] hover:bg-[#155d29] transition-all duration-300 flex items-center  
  justify-center"
```

```
  aria-label="Ir a proyecto anterior"
```

```
  aria-controls="projects-slider"
```

```
>
```

```
  <ChevronLeft className="w-6 h-6 text-gray-700 hover:text-white" />
```

```
</button>
```

```
<button
```

```
  className="w-12 h-12 rounded-full bg-white border border-gray-200  
  hover:border-[#155d29] hover:bg-[#155d29] transition-all duration-300 flex items-center  
  justify-center"
```

```
  aria-label="Ir a proyecto siguiente"
```

```
  aria-controls="projects-slider"
```

```
>
```

```
  <ChevronRight className="w-6 h-6 text-gray-700 hover:text-white" />
```

```
</button>
```

```
// Botones de navegación de slides
```

```
<button  
  className="w-3 h-3 rounded-full transition-all duration-200 bg-gray-300  
  hover:bg-gray-400"  
  
  aria-label={`Ir a proyecto ${index + 1}`}  
  
  aria-current={currentIndex === index ? 'true' : 'false'}  
  
  onClick={() => goToSlide(index)}  
  
>  
  
</button>  
  
```
```

### ### 6.2 Mejorar Contraste de Colores

```
```.css  
  
/* tailwind.config.js */  
  
module.exports = {  
  
  theme: {  
  
    extend: {  
  
      colors: {  
  
        'transervica-green': {  
  
          50: '#f0f9f1',  
  
          100: '#d1e8d4',  
  
          200: '#a3d1a8',  
  
          300: '#6fb578',  
  
        },  
  
      },  
  
    },  
  
  },  
  
}
```

```
400: '#4a9a52',
500: '#2d7d38',    // Original: #155d29
600: '#22662d',    // Nuevo para mejor contraste
700: '#1a5123',
800: '#15411c',
900: '#0d2e13',
},
},
},
},
};
...

```

****Botón con contraste mejorado:****

```
```jsx
<button
 className="bg-transervica-green-600 hover:bg-transervica-green-700 text-white px-6
py-3 rounded-md text-sm font-semibold shadow-md hover:shadow-lg transition-all
duration-300 focus:outline-none focus:ring-2 focus:ring-transervica-green-500
focus:ring-offset-2"
 aria-label="Solicitar cotización"
>
 QUOTE
</button>

```

...

### ### 6.3 Touch Targets Mínimos 48x48px

```css

/* Aumentar tamaño de botones táctiles */

.touch-target {

min-width: 48px;

min-height: 48px;

padding: 12px 16px;

}

/* Botones de navegación de slider */

.slider-nav-button {

width: 48px !important;

height: 48px !important;

min-width: 48px;

min-height: 48px;

}

/* Botones de paginación */

.pagination-dot {

width: 12px !important;

```
height: 12px !important;

min-width: 48px;

min-height: 48px;

padding: 18px;

border-radius: 50%;

}

...

---
```

🔒 FASE 7: SEGURIDAD CSP (Semana 3)

7.1 Implementar CSP con Nonces

```
```.javascript

// middleware/csp.js

const crypto = require('crypto');

function generateNonce() {

 return crypto.randomBytes(16).toString('base64');

}

function cspMiddleware(req, res, next) {
```



```
const nonce = generateNonce();

req.nonce = nonce;

const cspHeader = `
 default-src 'self';

 script-src 'self' 'nonce-${nonce}' 'strict-dynamic' https;;
 style-src 'self' 'nonce-${nonce}' https://fonts.googleapis.com;
 font-src 'self' https://fonts.gstatic.com;
 img-src 'self' data: https;;
 connect-src 'self' https;;
 frame-src https://www.youtube.com;
 object-src 'none';
 base-uri 'self';
 form-action 'self';
 require-trusted-types-for 'script';
 `.replace(/\\s+/g, ' ').trim();

res.setHeader('Content-Security-Policy', cspHeader);
res.setHeader('X-Content-Type-Options', 'nosniff');
res.setHeader('X-Frame-Options', 'DENY');
res.setHeader('X-XSS-Protection', '1; mode=block');

next();
```

```
}
```

```
module.exports = cspMiddleware;
```

```
...
```

**\*\*Uso en templates:\*\***

```
``jsx
```

```
// _document.jsx
```

```
import { Head, Html, Main, NextScript } from 'next/document';
```

```
export default function Document() {
```

```
 const nonce = process.env.NODE_ENV === 'production'
```

```
 ? crypto.randomBytes(16).toString('base64')
```

```
 : '';
```

```
 return (
```

```
 <Html>
```

```
 <Head>
```

```
 <meta httpEquiv="Content-Security-Policy" content={`
```

```
 default-src 'self';
```

```
 script-src 'self' 'nonce-${nonce}' 'strict-dynamic' https;;
```

```
 style-src 'self' 'nonce-${nonce}' https://fonts.googleapis.com;
```

```
 font-src 'self' https://fonts.gstatic.com;
```

```

img-src 'self' data: https;;

connect-src 'self' https;;

frame-src https://www.youtube.com;

object-src 'none';

base-uri 'self';

form-action 'self';

require-trusted-types-for 'script';

`.replace(/s+/g, ' ') />

</Head>

<body>

 <Main />

 <NextScript nonce={nonce} />

</body>

</Html>

);

}

...

📊 FASE 8: MONITOREO Y PRUEBAS (Semana 4)

8.1 Script de Pruebas Automatizadas

```

```
````javascript

// scripts/test-performance.js

const lighthouse = require('lighthouse');
const chromeLauncher = require('chrome-launcher');
const { writeFileSync } = require('fs');

async function runLighthouse(url, device = 'mobile') {

  const chrome = await chromeLauncher.launch({ chromeFlags: ['--headless'] });

  const options = {

    logLevel: 'info',

    output: 'json',

    onlyCategories: ['performance', 'accessibility', 'best-practices', 'seo'],

    port: chrome.port,

    formFactor: device,

    screenEmulation: device === 'mobile'

      ? { width: 375, height: 667, deviceScaleFactor: 2, mobile: true }

      : { width: 1350, height: 940, deviceScaleFactor: 1, mobile: false },

  };

  const runnerResult = await lighthouse(url, options);

  await chrome.kill();
```

```

const { scores, metrics } = runnerResult.lhr;

console.log(`\n=== Resultados ${device.toUpperCase()} ===`);
console.log(`Performance: ${Math.round(scores.performance * 100)}%`);
console.log(`Accessibility: ${Math.round(scores.accessibility * 100)}%`);
console.log(`Best Practices: ${Math.round(scores['best-practices'] * 100)}%`);
console.log(`SEO: ${Math.round(scores.seo * 100)}%`);

console.log(`\nMétricas Core Web Vitals:`);
console.log(`FCP: ${metrics.firstContentfulPaint.value.toFixed(2)}s`);
console.log(`LCP: ${metrics.largestContentfulPaint.value.toFixed(2)}s`);
console.log(`TBT: ${metrics.totalBlockingTime.value.toFixed(0)}ms`);
console.log(`CLS: ${metrics.cumulativeLayoutShift.value.toFixed(3)}%`);

writeFileSync(`lighthouse-${device}.json`, JSON.stringify(runnerResult.lhr, null, 2));

return scores.performance;
}

// Ejecutar pruebas
(async () => {
  const mobileScore = await runLighthouse('https://grupotranservica.com', 'mobile');

```

```
const desktopScore = await runLighthouse('https://grupotranservica.com', 'desktop');

if (mobileScore >= 0.95 && desktopScore >= 0.95) {
  console.log('\n✅ ¡Objetivo alcanzado! Puntuación >95 en ambos dispositivos');
} else {
  console.log('\n⚠️ Se necesitan más optimizaciones');
}
})();
...

```

8.2 Checklist Final

```markdown

## ## ✅ CHECKLIST DE IMPLEMENTACIÓN

### ### Imágenes (Semana 1)

- [ ] Optimizar imagen hero principal (5 versiones responsive)
- [ ] Optimizar logo (3 versiones)
- [ ] Optimizar todas las imágenes de proyectos
- [ ] Implementar <picture> con srcset
- [ ] Añadir fetchpriority="high" a LCP
- [ ] Eliminar lazy-loading de imagen principal

### ### YouTube (Semana 1-2)

- [ ] Crear componente LiteYouTubeEmbed
- [ ] Reemplazar iframes por lazy-load
- [ ] Precargar thumbnails
- [ ] Eliminar scripts de YouTube de carga inicial

### ### JavaScript (Semana 2)

- [ ] Implementar code splitting por rutas
- [ ] Eliminar código no usado (tree shaking)
- [ ] Diferir scripts no críticos
- [ ] Minificar y comprimir bundles
- [ ] Dividir vendor chunks

### ### CSS (Semana 2)

- [ ] Extraer CSS crítico (<10KB)
- [ ] Inlinar CSS crítico
- [ ] Diferir CSS no crítico
- [ ] Eliminar estilos no usados (PurgeCSS)

### ### Caché (Semana 3)

- [ ] Configurar headers HTTP (nginx/apache)
- [ ] Implementar Service Worker
- [ ] Configurar precaching de assets críticos

- [ ] Validar estrategias de caché

### ### Accesibilidad (Semana 3)

- [ ] Añadir aria-label a todos los botones
- [ ] Mejorar contraste de colores (ratio 4.5:1)
- [ ] Aumentar touch targets a 48x48px
- [ ] Añadir skip links
- [ ] Validar con axe-core

### ### Seguridad (Semana 3)

- [ ] Implementar CSP con nonces
- [ ] Añadir Trusted Types
- [ ] Configurar headers de seguridad
- [ ] Eliminar 'unsafe-inline'

### ### Pruebas (Semana 4)

- [ ] Ejecutar Lighthouse mobile
- [ ] Ejecutar Lighthouse desktop
- [ ] Probar en 5 dispositivos reales
- [ ] Validar Core Web Vitals
- [ ] Pruebas de carga en 3G lenta

...



---

## ## RESULTADOS ESPERADOS

| Métrica                           | Antes    | Después   | Mejora |
|-----------------------------------|----------|-----------|--------|
| -----                             | -----    | -----     | -----  |
| <b>**Rendimiento Mobile**</b>     | 56       | 95+       | +39    |
| <b>**Rendimiento Desktop**</b>    | 71       | 98+       | +27    |
| <b>**FCP Mobile**</b>             | 2.8s     | <1.2s     | -57%   |
| <b>**LCP Mobile**</b>             | 6.6s     | <1.8s     | -73%   |
| <b>**TBT Mobile**</b>             | 440ms    | <150ms    | -66%   |
| <b>**Tamaño Total**</b>           | 3031 KiB | <1200 KiB | -60%   |
| <b>**Accesibilidad**</b>          | 88       | 95+       | +7     |
| <b>**Prácticas Recomendadas**</b> | 69       | 95+       | +26    |

---

## ## COMANDOS RÁPIDOS

```
```bash
```

```
# 1. Optimizar todas las imágenes
```

```
npm run optimize:images
```

2. Build optimizado

npm run build:optimized

3. Ejecutar pruebas de rendimiento

npm run test:performance

4. Validar accesibilidad

npm run test:accessibility

5. Deploy con optimizaciones

npm run deploy:optimized

...

Voy a profundizar al máximo en cada área de optimización, proporcionando implementaciones completas, casos de estudio y técnicas avanzadas para llevar TRANSERVICA al 100% de rendimiento.

 GUÍA COMPLETA DE OPTIMIZACIÓN WEB - TRANSERVICA

De 56 a 100 puntos: Implementación Definitiva

 PARTE 1: ARQUITECTURA DE RENDIMIENTO

1.1 Análisis de Carga Crítica (Critical Rendering Path)

El navegador sigue este proceso para renderizar tu página:

...

HTML → Parse HTML → DOM



CSS → Parse CSS → CSSOM



JavaScript (bloqueante)



DOM + CSSOM = Render Tree



Layout (reflow) → Paint → Composite

...

****Problema actual en TRANSERVICA:****

- El CSS bloquea el renderizado por 310ms
- JavaScript de YouTube bloquea el hilo principal por 936ms
- 23 cadenas de dependencia críticas (máximo recomendado: 3)

1.2 Implementación de Carga Progresiva

```
```html
```

```
<!DOCTYPE html>
```

```
<html lang="es" dir="ltr">
```

```
<head>
```

```
<!-- Metadatos críticos primero -->
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0,
viewport-fit=cover">
```

```
<meta name="description" content="TRANSERVICA - Transporte de cargas excepcionales en
Venezuela hasta 1100 toneladas. Soluciones logísticas especializadas.">
```

```
<!-- Preconexiones estratégicas -->
```

```
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

```
<link rel="preconnect" href="https://grupotranservica.com" crossorigin>
```

```
<link rel="dns-prefetch" href="https://www.youtube.com">
```

```
<link rel="dns-prefetch" href="https://static.doubleclick.net">
```

```
<link rel="dns-prefetch" href="https://googleads.g.doubleclick.net">
```

```
<!-- Precarga de recursos críticos con prioridad -->
```

```
<link rel="preload" href="/assets/logo-transervica-200.webp" as="image"
type="image/webp" fetchpriority="high">
```

```
<link rel="preload" href="/fonts/inter-400.woff2" as="font" type="font/woff2" crossorigin>
```

```
<link rel="preload" href="/assets/hero-bg-800.webp" as="image" type="image/webp"
fetchpriority="high" media="(min-width: 801px)">
```

```
<link rel="preload" href="/assets/hero-bg-400.webp" as="image" type="image/webp"
fetchpriority="high" media="(max-width: 800px)">
```

```
<!-- CSS Crítico Inline (14KB máximo) -->
```

```
<style>
```

```
/* CSS Reset mínimo crítico */
```

```
,::before,*::after{box-sizing:border-box;margin:0;padding:0}
```

```
html{scroll-behavior:smooth;-webkit-text-size-adjust:100%}
```

```
body{font-family:Inter,system-ui,-apple-system,sans-serif;line-height:1.5;color:#1a1a1a;background-color:#fff;overflow-x:hidden}
```

```
img{max-width:100%;height:auto;display:block}
```

```
/* Variables de diseño */
```

```
:root{
```

```
--primary:#155d29;
```

```
--primary-dark:#0d3d1a;
```

```
--secondary:#3ea30f;
```

```
--accent:#f59e0b;
```

```
--text:#1f2937;
```

```
--text-light:#6b7280;
```

```
--bg:#ffffff;
```

```
--bg-alt:#f3f4f6;
```

```
--header-height:72px;
```

```
--section-padding:clamp(3rem,8vw,6rem)
}
```

```
/* Header crítico - siempre visible */
```

```
.site-header{

 position:fixed;

 top:0;

 left:0;

 right:0;

 height:var(--header-height);

 background:rgba(255,255,255,0.95);

 backdrop-filter:blur(10px);

 z-index:1000;

 border-bottom:1px solid rgba(0,0,0,0.05);

 transform:translateZ(0);

 will-change:transform
}
```

```
.header-container{

 max-width:1280px;

 margin:0 auto;

 padding:0 1rem;

 height:100%;

 display:flex;
```

```
 align-items:center;

 justify-content:space-between
}

.logo{

 height:48px;

 width:auto;

 display:block;

 content-visibility:auto

}

@media(min-width:640px){.logo{height:56px}}

@media(min-width:1024px){.logo{height:64px}}
```

```
/* Navegación móvil crítica */
```

```
.nav-toggle{

 display:flex;

 flex-direction:column;

 gap:5px;

 padding:12px;

 background:none;

 border:none;

 cursor:pointer;

 min-height:48px;

 min-width:48px
```

```
}
```

```
.nav-toggle span{
```

```
 display:block;
```

```
 width:24px;
```

```
 height:2px;
```

```
 background:var(--primary);
```

```
 transition:0.3s;
```

```
 transform-origin:center
```

```
}
```

```
@media(min-width:768px){.nav-toggle{display:none}}
```

```
/* Hero section - LCP crítico */
```

```
.hero{
```

```
 min-height:100vh;
```

```
 min-height:100dvh;
```

```
 display:flex;
```

```
 align-items:center;
```

```
 justify-content:center;
```

```
 position:relative;
```

```
 padding-top:var(--header-height);
```

```
 background:linear-gradient(135deg,var(--primary)0%,var(--primary-dark)100%);
```

```
 overflow:hidden;
```

```
 contain:layout style paint
```



```
}

.hero-bg{

 position:absolute;

 inset:0;

 opacity:0.15;

 object-fit:cover;

 width:100%;

 height:100%;

 pointer-events:none;

 transform:translateZ(0)

}

.hero-content{

 position:relative;

 z-index:2;

 text-align:center;

 padding:2rem;

 max-width:800px;

 color:#fff;

 animation:fadeInUp 0.8s ease-out

}

@keyframes fadeInUp{

 from{opacity:0;transform:translateY(30px)}

 to{opacity:1;transform:translateY(0)}
```

```
}
```

```
.hero h1{
```

```
font-size:clamp(2rem,5vw,3.5rem);
```

```
font-weight:700;
```

```
line-height:1.1;
```

```
margin-bottom:1.5rem;
```

```
text-wrap:balance
```

```
}
```

```
.hero p{
```

```
font-size:clamp(1rem,2vw,1.25rem);
```

```
opacity:0.9;
```

```
margin-bottom:2rem;
```

```
max-width:600px;
```

```
margin-left:auto;
```

```
margin-right:auto
```

```
}
```

```
.btn-primary{
```

```
display:inline-flex;
```

```
align-items:center;
```

```
gap:0.5rem;
```

```
padding:1rem 2rem;
```

```
background:var(--accent);
```

```
color:#1a1a1a;
```

```
font-weight:600;

text-decoration:none;

border-radius:0.5rem;

transition:transform 0.2s,box-shadow 0.2s;

min-height:48px;

border:none;

cursor:pointer;

font-size:1rem;

will-change:transform
}
```

```
.btn-primary:hover{

 transform:translateY(-2px);

 box-shadow:0 10px 20px rgba(0,0,0,0.2)

}
```

```
.btn-primary:active{transform:translateY(0)}
```

```
/* Estados de carga */
```

```
.skeleton{

 background:linear-gradient(90deg,#f0f0f0 25%,#e0e0e0 50%,#f0f0f0 75%);

 background-size:200% 100%;

 animation:loading 1.5s infinite;

 border-radius:4px

}
```

```
@keyframes loading{
 0%{background-position:200% 0}
 100%{background-position:-200% 0}
}
```

```
/* Utilidades de accesibilidad */
```

```
.sr-only{
 position:absolute;
 width:1px;
 height:1px;
 padding:0;
 margin:-1px;
 overflow:hidden;
 clip:rect(0,0,0,0);
 white-space:nowrap;
 border-width:0
}
```

```
/* Contenido diferido - oculto inicialmente */
```

```
.deferred-section{
 content-visibility:auto;
 contain-intrinsic-size:0 500px
}
```

</style>

<!-- CSS no crítico cargado asíncronamente -->

<link rel="preload" href="/assets/styles/async.css" as="style"  
onload="this.onload=null;this.rel='stylesheet'">

<noscript><link rel="stylesheet" href="/assets/styles/async.css"></noscript>

<!-- Fuentes con display=swap para evitar FOIT -->

<link rel="stylesheet"  
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap"  
media="print" onload="this.media='all'">

<noscript><link rel="stylesheet"  
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap"  
></noscript>

<title>TRANSERVICA | Transporte de Cargas Excepcionales Venezuela</title>

</head>

<body>

<!-- Header -->

<header class="site-header">

<div class="header-container">

<a href="/" aria-label="TRANSERVICA - Inicio">



</a>

<button class="nav-toggle" aria-label="Abrir menú de navegación" aria-expanded="false"
aria-controls="main-nav">

    <span></span>

    <span></span>

    <span></span>

</button>

<!-- Nav desktop cargado diferido -->

<nav id="main-nav" class="main-nav" style="display:none;">

    <!-- Contenido cargado vía JS -->

</nav>

</div>

</header>


<!-- Hero - Contenido crítico above the fold -->

<section class="hero" id="inicio">

    
<div class="hero-content">
    <h1>Transporte de Cargas Excepcionales en Venezuela</h1>
    <p>Soluciones logísticas especializadas para cargas hasta 1100 toneladas. Experiencia,
tecnología y seguridad garantizada.</p>
    <a href="#contacto" class="btn-primary">
        Solicitar Cotización
        <svg width="20" height="20" viewBox="0 0 20 20" fill="none" aria-hidden="true">
            <path d="M4.167 10h11.666m0 0L10 4.167M15.833 10L10 15.833" stroke="currentColor"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"/>
        </svg>
    </a>
```

```
</div>
```

```
</section>
```

```
<!-- Contenido diferido -->
```

```
<main>
```

```
<section class="deferred-section" id="servicios" data-section="services">
```

```
<!-- Cargado vía Intersection Observer -->
```

```
</section>
```

```
<section class="deferred-section" id="proyectos" data-section="projects">
```

```
<!-- Cargado vía Intersection Observer -->
```

```
</section>
```

```
<section class="deferred-section" id="equipos" data-section="equipment">
```

```
<!-- Cargado vía Intersection Observer -->
```

```
</section>
```

```
<section class="deferred-section" id="contacto" data-section="contact">
```

```
<!-- Cargado vía Intersection Observer -->
```

```
</section>
```

```
</main>
```

```
<!-- Scripts diferidos -->
```

```
<script>
```

```
// Critical JavaScript inline (< 3KB)
```

```
(function(){
```



```
'use strict';

// Detección de soporte para lazy loading
if('loading' in HTMLImageElement.prototype){
  document.documentElement.classList.add('native-lazy-loading');
}

// Precarga de secciones visibles
const observerOptions = {
  rootMargin: '200px 0px',
  threshold: 0.01
};

const sectionObserver = new IntersectionObserver((entries) => {
  entries.forEach(entry => {
    if(entry.isIntersecting){
      const section = entry.target;

      const sectionName = section.dataset.section;

      loadSection(sectionName, section);

      sectionObserver.unobserve(section);
    }
  });
}, observerOptions);
```

```
document.querySelectorAll('.deferred-section').forEach(section => {  
  
  sectionObserver.observe(section);  
  
});  
  
// Función de carga de secciones  
  
function loadSection(name, container){  
  
  // Marcar como cargando  
  
  container.innerHTML = '<div class="skeleton" style="height:400px"></div>';  
  
  
  // Simular carga real (reemplazar con fetch real)  
  
  import(`/assets/sections/${name}.js`)  
  
    .then(module => {  
  
      module.render(container);  
  
    })  
  
    .catch(() => {  
  
      // Fallback: cargar HTML estático  
  
      fetch(`/assets/sections/${name}.html`)  
  
        .then(r => r.text())  
  
        .then(html => {  
  
          container.innerHTML = html;  
  
          // Ejecutar scripts inline  
  
          const scripts = container.querySelectorAll('script');
```

```
scripts.forEach(script => {  
  
  const newScript = document.createElement('script');  
  
  newScript.textContent = script.textContent;  
  
  document.head.appendChild(newScript);  
  
  });  
  
  });  
  
});  
}
```

// Menú móvil

```
const navToggle = document.querySelector('.nav-toggle');
```

```
let navLoaded = false;
```

```
navToggle.addEventListener('click', () => {
```

```
  if(!navLoaded){
```

```
    // Cargar navegación bajo demanda
```

```
    import('/assets/nav-mobile.js').then(m => m.init());
```

```
    navLoaded = true;
```

```
  }
```

```
  const expanded = navToggle.getAttribute('aria-expanded') === 'true';
```

```
  navToggle.setAttribute('aria-expanded', !expanded);
```

```
});
```

```
})();
```

```
</script>
```

```
<!-- Service Worker -->
```

```
<script>
```

```
  if('serviceWorker' in navigator){  
    window.addEventListener('load', () => {  
      navigator.serviceWorker.register('/sw.js');  
    });  
  }
```

```
</script>
```

```
</body>
```

```
</html>
```

```
...
```

```
---
```

PARTE 2: SISTEMA AVANZADO DE IMÁGENES

2.1 Pipeline de Optimización de Imágenes

```
```javascript
```

```
// scripts/optimize-images.js
```

```
const sharp = require('sharp');
```

```
const fs = require('fs').promises;
```

```
const path = require('path');
```

```
const glob = require('glob');
```

```
const CONFIG = {
```

```
 inputDir: './src/assets/images',
```

```
 outputDir: './public/assets',
```

```
 formats: ['webp', 'avif'],
```

```
 sizes: [400, 800, 1200, 1920],
```

```
 quality: {
```

```
 webp: 80,
```

```
 avif: 75,
```

```
 jpeg: 85
```

```
 }
```

```
};
```

```
async function optimizeImage(inputPath) {
```

```
 const filename = path.basename(inputPath, path.extname(inputPath));
```

```
 const ext = path.extname(inputPath).toLowerCase();
```

```
 // Leer imagen original
```

```
 const image = sharp(inputPath);
```

```
 const metadata = await image.metadata();
```

```
const results = [];

// Generar versiones responsivas
for (const width of CONFIG.sizes) {
 if (width > metadata.width) continue;

 const resizeOptions = {
 width,
 withoutEnlargement: true,
 fit: 'inside'
 };

 // WebP
 const webpPath = `${CONFIG.outputDir}/${filename}-${width}.webp`;
 await image
 .clone()
 .resize(resizeOptions)
 .webp({
 quality: CONFIG.quality.webp,
 effort: 6,
 smartSubsample: true
 })
```

```
.ToFile(webpPath);

results.push({
 width,
 format: 'webp',
 path: webpPath,
 size: (await fs.stat(webpPath)).size
});

// AVIF para navegadores modernos
const avifPath = `${CONFIG.outputDir}/${filename}-${width}.avif`;

await image
 .clone()
 .resize(resizeOptions)
 .avif({
 quality: CONFIG.quality.avif,
 effort: 4,
 chromaSubsampling: '4:2:0'
 })
 .ToFile(avifPath);

results.push({
 width,
```

```
 format: 'avif',
 path: avifPath,
 size: (await fs.stat(avifPath)).size
 });
}
```

```
// Generar placeholder LQIP (Low Quality Image Placeholder)
```

```
const lqipPath = `${CONFIG.outputDir}/${filename}-lqip.jpg`;
```

```
await image
```

```
 .clone()
```

```
 .resize(20, null, { withoutEnlargement: true })
```

```
 .jpeg({ quality: 20, progressive: true })
```

```
 .ToFile(lqipPath);
```

```
// Generar blurhash o base64 para placeholder
```

```
const lqipBuffer = await fs.readFile(lqipPath);
```

```
const base64 = `data:image/jpeg;base64,${lqipBuffer.toString('base64')}`;
```

```
return {
```

```
 filename,
```

```
 originalWidth: metadata.width,
```

```
 originalHeight: metadata.height,
```

```
 variants: results,
```



```
 placeholder: base64
 };
}
```

// Componente React/Vue para imágenes optimizadas

// components/OptimizedImage.jsx

```
import { useState, useEffect, useRef } from 'react';
```

```
export function OptimizedImage({
 src,
 alt,
 width,
 height,
 className = '',
 sizes = '100vw',
 priority = false,
 objectFit = 'cover'
}) {
 const [isLoading, setIsLoaded] = useState(false);
 const [isInView, setIsInView] = useState(false);
 const imgRef = useRef(null);

 useEffect(() => {
```

```
if (priority) {
 setIsInView(true);
 return;
}
```

```
const observer = new IntersectionObserver(
 ([entry]) => {
 if (entry.isIntersecting) {
 setIsInView(true);
 observer.disconnect();
 }
 },
 { rootMargin: '200px' }
);
```

```
if (imgRef.current) {
 observer.observe(imgRef.current);
}
```

```
return () => observer.disconnect();
}, [priority]);
```

```
// Generar srcset basado en convención de nombres
```

```
const generateSrcSet = (format) => {
 return [400, 800, 1200, 1920]
 .filter(w => w <= (width || 1920))
 .map(w => `/assets/${src}-${w}.${format} ${w}w`)
 .join(', ');
};
```

```
return (
 <div
 ref={imgRef}
 className={`optimized-image-container ${className}`}
 style={{
 aspectRatio: `${width}/${height}`,
 backgroundColor: '#f0f0f0',
 overflow: 'hidden'
 }}
 >
 {isInView && (
 <picture>
 {/* AVIF para navegadores modernos */}
 <source
 type="image/avif"
 srcSet={generateSrcSet('avif')}
 />
)
)
 }
```

```
 sizes={sizes}

 />

 { /* WebP como fallback moderno */ }

 <source

 type="image/webp"

 srcSet={generateSrcSet('webp')}

 sizes={sizes}

 />

 { /* JPEG/PNG como último recurso */ }

 <img

 src={` /assets/${src}-800.jpg`}

 srcSet={generateSrcSet('jpg')}

 sizes={sizes}

 alt={alt}

 width={width}

 height={height}

 loading={priority ? 'eager' : 'lazy'}

 decoding={priority ? 'sync' : 'async'}

 fetchpriority={priority ? 'high' : 'auto'}

 onLoad={() => setIsLoaded(true)}

 style={{

 width: '100%',

 height: '100%',
```

```

 objectFit,

 opacity: isLoading ? 1 : 0,

 transition: 'opacity 0.3s ease-in-out',

 transform: 'translateZ(0)' // GPU acceleration
 }}

 />

</picture>

)}

{!isLoading && (

 <div

 className="image-placeholder"

 style={{

 position: 'absolute',

 inset: 0,

 filter: 'blur(20px)',

 transform: 'scale(1.1)',

 backgroundImage: `url(/assets/${src}-lqip.jpg)`,

 backgroundSize: 'cover',

 backgroundPosition: 'center'

 }}

 />

)}

</div>

```

```
);
}
...
```

### ### 2.2 Optimización Específica del Logo

```
```javascript
```

```
// El logo es el LCP en móvil - requiere tratamiento especial
```

```
// scripts/optimize-logo.js
```

```
const sharp = require('sharp');
```

```
async function optimizeLogo() {
```

```
  const input = './src/assets/logo-transervica.png';
```

```
  // El logo original es 1738x1482, lo necesitamos en múltiples tamaños
```

```
  const sizes = [
```

```
    { width: 128, height: 109, name: 'logo-transervica-128' }, // Móvil header
```

```
    { width: 160, height: 137, name: 'logo-transervica-160' }, // Móvil grande
```

```
    { width: 200, height: 171, name: 'logo-transervica-200' }, // Desktop header
```

```
    { width: 256, height: 219, name: 'logo-transervica-256' }, // Footer/hero
```

```
    { width: 400, height: 341, name: 'logo-transervica-400' } // Hero grande
```

```
];
```

```
for (const size of sizes) {
```

```
  // WebP con calidad alta para el logo (importante para marca)
```

```
  await sharp(input)
```

```
    .resize(size.width, size.height, {
```

```
      fit: 'contain',
```

```
      background: { r: 0, g: 0, b: 0, alpha: 0 }
```

```
    })
```

```
    .webp({
```

```
      quality: 90,
```

```
      alphaQuality: 100,
```

```
      lossless: size.width <= 200 // Sin pérdida para tamaños pequeños
```

```
    })
```

```
    .toFile(`./public/assets/${size.name}.webp`);
```

```
  // PNG fallback para transparencia perfecta
```

```
  await sharp(input)
```

```
    .resize(size.width, size.height, {
```

```
      fit: 'contain',
```

```
      background: { r: 0, g: 0, b: 0, alpha: 0 }
```

```
    })
```

```
    .png({
```

```

        quality: 95,
        compressionLevel: 9,
        adaptiveFiltering: true
    })

    .ToFile(`./public/assets/${size.name}.png`);
}

console.log('Logo optimizado en todas las resoluciones');
}

optimizeLogo();
...

---
```

🧑‍💻 PARTE 3: SISTEMA DE VIDEO OPTIMIZADO

3.1 Implementación Lite-YouTube Avanzada

```

```javascript
// components/LiteYouTube.jsx

import { useState, useRef, useEffect } from 'react';
```



```
const thumbnailCache = new Map();

export function LiteYouTube({
 videoid,
 title,
 startTime = 0,
 autoplay = false,
 className = '',
 aspectRatio = '16/9'
}) {
 const [isLoading, setIsLoaded] = useState(false);
 const [isPreconnected, setIsPreconnected] = useState(false);
 const [thumbnailUrl, setThumbnailUrl] = useState('');
 const containerRef = useRef(null);

 // Precargar thumbnail de alta calidad
 useEffect(() => {
 const loadThumbnail = async () => {
 if (thumbnailCache.has(videoid)) {
 setThumbnailUrl(thumbnailCache.get(videoid));
 return;
 }
 }
 })
}
```

```

// Intentar obtener thumbnail de máxima calidad

const qualities = ['maxresdefault', 'sddefault', 'hqdefault', 'mqdefault', 'default'];

for (const quality of qualities) {

 const url = `https://img.youtube.com/vi/${videoid}/${quality}.jpg`;

 try {

 const response = await fetch(url, { method: 'HEAD' });

 if (response.ok) {

 thumbnailCache.set(videoid, url);

 setThumbnailUrl(url);

 return;

 }

 } catch (e) {

 continue;

 }

}

};

loadThumbnail();

}, [videoid]);

// Precargar conexión a YouTube al hacer hover

const handleMouseEnter = () => {

```

```
if (!isPreconnected) {
 const link = document.createElement('link');
 link.rel = 'preconnect';
 link.href = 'https://www.youtube-nocookie.com';
 document.head.appendChild(link);
 setIsPreconnected(true);
}
};
```

```
const loadVideo = () => {
 setIsLoaded(true);
};
```

// URL del video con parámetros optimizados

```
const getVideoUrl = () => {
 const params = new URLSearchParams({
 autoplay: '1',
 start: startTime.toString(),
 rel: '0', // No mostrar videos relacionados
 modestbranding: '1', // Minimal branding
 playsinline: '1' // Reproducir inline en iOS
 });

 return `https://www.youtube-nocookie.com/embed/${videoid}?${params}`;
};
```

```
};
```

```
return (
```

```
<div
```

```
 ref={containerRef}
```

```
 className={`lite-youtube ${className}`} `
```

```
 style={{
```

```
 aspectRatio,
```

```
 backgroundColor: '#000',
```

```
 position: 'relative',
```

```
 overflow: 'hidden',
```

```
 borderRadius: '8px'
```

```
 }}
```

```
 onMouseEnter={handleMouseEnter}
```

```
 onClick={loadVideo}
```

```
 role="button"
```

```
 tabIndex={0}
```

```
 aria-label={`Reproducir video: ${title}`} `
```

```
 onKeyDown={(e) => e.key === 'Enter' && loadVideo() }
```

```
>
```

```
{!isLoading ? (
```

```
<>
```

```
 { /* Thumbnail optimizado */ }
```

```
{thumbnailUrl && (

 <img

 src={thumbnailUrl}

 alt=""

 loading="lazy"

 decoding="async"

 style={{

 position: 'absolute',

 inset: 0,

 width: '100%',

 height: '100%',

 objectFit: 'cover',

 filter: 'brightness(0.9)',

 transform: 'scale(1.05)',

 transition: 'transform 0.3s ease'

 }}

 onLoad={(e) => {

 e.target.style.transform = 'scale(1)';

 }}

 />

)}

{/* Overlay con gradiente */}
```

```
<div style={{
 position: 'absolute',
 inset: 0,
 background: 'linear-gradient(to top, rgba(0,0,0,0.6) 0%, transparent 60%)',
 pointerEvents: 'none'
}} />
```

```
{/* Título del video */}
```

```
<div style={{
 position: 'absolute',
 bottom: '16px',
 left: '16px',
 right: '16px',
 color: 'white',
 fontSize: '14px',
 fontWeight: 500,
 textShadow: '0 2px 4px rgba(0,0,0,0.5)',
 pointerEvents: 'none'
}}>
```

```
{title}
```

```
</div>
```

```
{/* Botón de play optimizado */}
```

```
<div style={{
 position: 'absolute',
 top: '50%',
 left: '50%',
 transform: 'translate(-50%, -50%)',
 width: '68px',
 height: '48px',
 backgroundColor: 'rgba(0,0,0,0.8)',
 borderRadius: '8px',
 display: 'flex',
 alignItems: 'center',
 justifyContent: 'center',
 transition: 'all 0.2s ease',
 cursor: 'pointer'
}} className="play-button">
```

```
<svg
 width="24"
 height="24"
 viewBox="0 0 24 24"
 fill="white"
 style={{ marginLeft: '4px' }}
>
 <path d="M8 5v14l11-7z"/>
```

```
</svg>
```

```
</div>
```

```
{/* Estilos hover */}
```

```
<style>{'
```

```
.lite-youtube:hover .play-button {
```

```
background-color: #ff0000 !important;
```

```
transform: translate(-50%, -50%) scale(1.1);
```

```
}
```

```
.lite-youtube:hover img {
```

```
transform: scale(1.1) !important;
```

```
}
```

```
'}</style>
```

```
</>
```

```
): (
```

```
<iframe
```

```
src={getVideoUrl()}
```

```
title={title}
```

```
allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope;
picture-in-picture"
```

```
allowFullScreen
```

```
style={{
```

```
position: 'absolute',
```



```
 inset: 0,

 width: '100%',

 height: '100%',

 border: 'none'

 }}

 />

)}

</div>

);

}
```

```
// Uso en página

// <LiteYouTube

// videoId="_LQbWkWlg6s"

// title="Proyecto Multicar Railway Convoy"

// startTime={120}

// />

...


```

### ### 3.2 Carga Progresiva de Videos (Intersection Observer)

```
```javascript

// hooks/useVideoLazyLoad.js
```

```
import { useEffect, useRef, useState } from 'react';

export function useVideoLazyLoad(options = {}) {

  const containerRef = useRef(null);

  const [shouldLoad, setShouldLoad] = useState(false);

  const [isNearViewport, setIsNearViewport] = useState(false);

  useEffect(() => {

    const container = containerRef.current;

    if (!container) return;

    // Precargar cuando está cerca (500px antes)

    const preloadObserver = new IntersectionObserver(

      ([entry]) => {

        if (entry.isIntersecting) {

          setIsNearViewport(true);

          preloadObserver.disconnect();

        }

      },

      { rootMargin: '500px' }

    );

    // Cargar real cuando es visible
```

```

const loadObserver = new IntersectionObserver(
  ([entry]) => {
    if (entry.isIntersecting) {
      setShouldLoad(true);
      loadObserver.disconnect();
    }
  },
  { rootMargin: '50px' }
);

preloadObserver.observe(container);
loadObserver.observe(container);

return () => {
  preloadObserver.disconnect();
  loadObserver.disconnect();
};
}, []);

return { containerRef, shouldLoad, isNearViewport };
}

// Implementación en grid de videos

```

```

// components/VideoGallery.jsx

export function VideoGallery({ videos }) {

  return (

    <div className="video-grid" style={{

      display: 'grid',

      gridTemplateColumns: 'repeat(auto-fit, minmax(300px, 1fr))',

      gap: '24px'

    }}>

      {videos.map((video, index) => (

        <VideoItem

          key={video.id}

          video={video}

          index={index}

        />

      ))}

    </div>

  );

}

function VideoItem({ video, index }) {

  const { containerRef, shouldLoad, isNearViewport } = useVideoLazyLoad();

  return (

```

```
<div ref={containerRef} style={{ minHeight: '200px' }}>
```

```
{isNearViewport && (
```

```
<>
```

```
  { /* Precargar thumbnail */ }
```

```
  <link
```

```
    rel="preload"
```

```
    as="image"
```

```
    href={`https://img.youtube.com/vi/${video.id}/hqdefault.jpg`}
```

```
  />
```

```
</>
```

```
)} 
```

```
{shouldLoad ? (
```

```
  <LiteYouTube
```

```
    videoId={video.id}
```

```
    title={video.title}
```

```
    startTime={video.startTime}
```

```
  />
```

```
): (
```

```
  <div className="video-skeleton" style={{
```

```
    aspectRatio: '16/9',
```

```
    background: 'linear-gradient(90deg, #f0f0f0 25%, #e0e0e0 50%, #f0f0f0 75%);',
```

```
    backgroundSize: '200% 100%',
```

```

        animation: 'shimmer 1.5s infinite',

        borderRadius: '8px'

    }} />

  )}

</div>

);

}

...

---
```

⚡ PARTE 4: OPTIMIZACIÓN DE JAVASCRIPT

4.1 Estrategia de Code Splitting Avanzada

```

````javascript
// vite.config.js

import { defineConfig } from 'vite';

import react from '@vitejs/plugin-react';

import { visualizer } from 'rollup-plugin-visualizer';

import compression from 'vite-plugin-compression';

export default defineConfig({
```

```
plugins: [
 react(),
 compression({
 algorithm: 'brotliCompress',
 ext: '.br',
 threshold: 1024,
 deleteOriginFile: false
 }),
 compression({
 algorithm: 'gzip',
 ext: '.gz',
 threshold: 1024
 }),
 visualizer({
 open: true,
 gzipSize: true,
 brotliSize: true,
 filename: 'dist/stats.html'
 })
],
```

```
build: {
 target: 'es2020',
```

```
cssTarget: 'chrome61',
```

```
rollupOptions: {
```

```
 output: {
```

```
 // Estrategia de chunking manual
```

```
 manualChunks(id) {
```

```
 // React y React DOM van juntos (siempre necesarios)
```

```
 if (id.includes('node_modules/react') || id.includes('node_modules/react-dom')) {
```

```
 return 'react-core';
```

```
 }
```

```
 // Router separado (cargado bajo demanda)
```

```
 if (id.includes('node_modules/react-router') || id.includes('node_modules/@remix-run'))
```

```
{
```

```
 return 'router';
```

```
 }
```

```
 // UI Components library (si usas una)
```

```
 if (id.includes('node_modules/@radix-ui') || id.includes('node_modules/@headlessui')) {
```

```
 return 'ui-components';
```

```
 }
```

```
 // Animaciones - solo cargadas cuando se necesitan
```



```
if (id.includes('node_modules/framer-motion') || id.includes('node_modules/gsap')) {
 return 'animations';
}

// Formularios y validación
if (id.includes('node_modules/react-hook-form') || id.includes('node_modules/zod')) {
 return 'forms';
}

// Mapas (carga pesada, solo en contacto)
if (id.includes('node_modules/leaflet') || id.includes('node_modules/mapbox')) {
 return 'maps';
}

// Utilidades compartidas
if (id.includes('node_modules/lodash') || id.includes('node_modules/date-fns')) {
 return 'utils';
}
},

// Nomenclatura de archivos para caché eficiente
entryFileNames: 'assets/[name]-[hash].js',
chunkFileNames: 'assets/[name]-[hash].js',
```

```
assetFileNames: (assetInfo) => {

 const info = assetInfo.name.split('.');

 const ext = info[info.length - 1];

 if (/^(.png|jpe?g|gif|svg|webp|avif)$/i.test(assetInfo.name)) {

 return 'assets/images/[name]-[hash][extname]';

 }

 if (ext === 'css') {

 return 'assets/styles/[name]-[hash][extname]';

 }

 if (ext === 'woff2' || ext === 'woff') {

 return 'assets/fonts/[name]-[hash][extname]';

 }

 return 'assets/[name]-[hash][extname]';

}

},
```

// Minificación avanzada

```
minify: 'terser',

terserOptions: {

 compress: {

 drop_console: true,
```

```
drop_debugger: true,

pure_funcs: ['console.log', 'console.info', 'console.debug'],

passes: 2,

dead_code: true,

unused: true

},

mangle: {

 safari10: true,

 properties: {

 regex: /^_/ // Solo manglear propiedades privadas

 }

},

format: {

 comments: false,

 ecma: 2020

}

},
```

```
// CSS
```

```
cssCodeSplit: true,

cssMinify: 'lightningcss',
```

```
// Reporte de tamaño
```

```

 reportCompressedSize: true,

 chunkSizeWarningLimit: 500
 },

 // Optimizaciones de dependencias
 optimizeDeps: {
 include: ['react', 'react-dom', 'react-router-dom'],
 exclude: ['@heavy-library/not-used']
 }
});
...

```

### ### 4.2 Sistema de Precarga Inteligente de Rutas

```

```javascript

// router/index.jsx

import { lazy, Suspense } from 'react';

import { createBrowserRouter, useLocation, useNavigation } from 'react-router-dom';

import { useEffect } from 'react';

// Componente de carga con skeleton específico

const PageLoader = ({ type }) => {

  const skeletons = {

```

```
    home: <HomeSkeleton />,
    projects: <ProjectsSkeleton />,
    contact: <ContactSkeleton />,
    default: <GenericSkeleton />
  };

  return skeletons[type] || skeletons.default;
};
```

// Lazy loading con prefetching

```
const HomePage = lazy(() => import('./pages/Home'));
const ProjectsPage = lazy(() => import('./pages/Projects'));
const ContactPage = lazy(() => import('./pages/Contact'));
const EquipmentPage = lazy(() => import('./pages/Equipment'));
```

// Precarga de rutas basada en navegación probable

```
const routePrefetchMap = {
  '/': ['Projects', 'Contact'],
  '/proyectos': ['Equipment', 'Contact'],
  '/equipos': ['Contact'],
  '/contacto': []
};
```

```
let prefetchTimeout;
```

```
function usePrefetch() {

  const location = useLocation();

  const navigation = useNavigation();

  useEffect(() => {

    const routesToPrefetch = routePrefetchMap[location.pathname] || [];

    // Limpiar timeout anterior

    clearTimeout(prefetchTimeout);

    // Esperar a que la página actual esté estable

    prefetchTimeout = setTimeout(() => {

      if (navigation.state === 'idle') {

        routesToPrefetch.forEach(routeName => {

          // Precargar componente

          import(`./pages/${routeName}.jsx`);

          // Precargar datos si es necesario

          if (routeName === 'Projects') {

            fetch('/api/projects-preview').catch(() => {});

          }

        });

      }

    });

  });

}
```

```

    }

    }, 2000); // Esperar 2s después de carga estable

    return () => clearTimeout(prefetchTimeout);

  }, [location.pathname, navigation.state]);
}

// Router con prefetching
export const router = createBrowserRouter([
  {
    path: '/',
    element: <Root />,
    children: [
      {
        index: true,
        element: (
          <Suspense fallback={<PageLoader type="home" />}>
            <HomePage />
          </Suspense>
        )
      },
      {
        path: 'proyectos',

```

```
element: (  
  <Suspense fallback={<PageLoader type="projects" />}>  
    <ProjectsPage />  
  </Suspense>  
)  
,  
loader: () => {  
  // Precargar imágenes de proyectos  
  return fetch('/api/projects').then(r => r.json());  
}  
,  
{  
  path: 'equipos',  
  element: (  
    <Suspense fallback={<PageLoader type="equipment" />}>  
      <EquipmentPage />  
    </Suspense>  
  )  
},  
{  
  path: 'contacto',  
  element: (  
    <Suspense fallback={<PageLoader type="contact" />}>  
      <ContactPage />  
    </Suspense>  
  )  
}
```



```

        </Suspense>
    )
}
]
}
]);

```

```

function Root() {
    usePrefetch();
    return <Outlet />;
}
...

```

4.3 Web Workers para Procesamiento Pesado

```

````javascript
// workers/imageProcessor.worker.js

self.addEventListener('message', async (e) => {
 const { type, imageData, operations } = e.data;

 switch (type) {
 case 'compress':
 const compressed = await compressImage(imageData, operations.quality);

```

```

 self.postMessage({ type: 'compressed', data: compressed }, [compressed]);

 break;

case 'generateBlurhash':

 const blurhash = await encodeBlurhash(imageData);

 self.postMessage({ type: 'blurhash', data: blurhash });

 break;

case 'resize':

 const resized = await resizeImage(imageData, operations.dimensions);

 self.postMessage({ type: 'resized', data: resized }, [resized]);

 break;

}

});

```

```

async function compressImage(imageData, quality) {

 // Implementación de compresión usando WebAssembly o algoritmos JS

 // Esto corre en thread separado, no bloquea UI

}

```

// Uso en componente

// hooks/useImageWorker.js

```
import { useEffect, useRef, useCallback } from 'react';
```

```
export function useImageWorker() {

 const workerRef = useRef(null);

 useEffect(() => {

 workerRef.current = new Worker(

 new URL('../workers/imageProcessor.worker.js', import.meta.url),

 { type: 'module' }

);

 return () => workerRef.current?.terminate();

 }, []);

 const processImage = useCallback((imageData, operations) => {

 return new Promise((resolve, reject) => {

 const worker = workerRef.current;

 const id = Math.random().toString(36).substr(2, 9);

 const handler = (e) => {

 if (e.data.id === id) {

 worker.removeEventListener('message', handler);

 if (e.data.error) reject(e.data.error);

 else resolve(e.data.result);

 }

 };

 });

 });

}
```

```

 }

 };

 worker.addEventListener('message', handler);

 worker.postMessage({ id, imageData, operations });

 });

 }, []);

 return { processImage };

}

...

```

## ## 🎨 PARTE 5: CSS Y ESTILOS OPTIMIZADOS

### ### 5.1 Sistema de Diseño con Container Queries

```

```css

/* styles/critical.css - 14KB máximo */

/* Reset moderno */

*,
```

```
*::before,  
*::after {  
  
  box-sizing: border-box;  
  
  margin: 0;  
  
  padding: 0;  
  
}
```

```
/* Propiedades personalizadas globales */
```

```
:root {
```

```
  /* Colores */
```

```
  --color-primary: #155d29;
```

```
  --color-primary-dark: #0d3d1a;
```

```
  --color-primary-light: #1a7a36;
```

```
  --color-secondary: #3ea30f;
```

```
  --color-accent: #f59e0b;
```

```
  --color-accent-hover: #d97706;
```

```
  /* Escala tipográfica fluida */
```

```
  --text-xs: clamp(0.75rem, 0.7rem + 0.25vw, 0.875rem);
```

```
  --text-sm: clamp(0.875rem, 0.8rem + 0.35vw, 1rem);
```

```
  --text-base: clamp(1rem, 0.95rem + 0.25vw, 1.125rem);
```

```
  --text-lg: clamp(1.125rem, 1rem + 0.5vw, 1.25rem);
```

```
  --text-xl: clamp(1.25rem, 1.1rem + 0.75vw, 1.5rem);
```

```
--text-2xl: clamp(1.5rem, 1.3rem + 1vw, 2rem);  
  
--text-3xl: clamp(1.875rem, 1.5rem + 1.5vw, 2.5rem);  
  
--text-4xl: clamp(2.25rem, 1.8rem + 2vw, 3.5rem);
```

/ Espaciado fluido */*

```
--space-xs: clamp(0.5rem, 0.4rem + 0.5vw, 0.75rem);  
  
--space-sm: clamp(0.75rem, 0.6rem + 0.75vw, 1rem);  
  
--space-md: clamp(1rem, 0.8rem + 1vw, 1.5rem);  
  
--space-lg: clamp(1.5rem, 1.2rem + 1.5vw, 2.5rem);  
  
--space-xl: clamp(2rem, 1.5rem + 2.5vw, 4rem);  
  
--space-2xl: clamp(3rem, 2rem + 5vw, 6rem);
```

/ Layout */*

```
--header-height: clamp(60px, 10vw, 80px);  
  
--container-max: 1280px;  
  
--container-padding: clamp(1rem, 5vw, 2rem);
```

/ Sombras */*

```
--shadow-sm: 0 1px 2px 0 rgb(0 0 0 / 0.05);  
  
--shadow-md: 0 4px 6px -1px rgb(0 0 0 / 0.1);  
  
--shadow-lg: 0 10px 15px -3px rgb(0 0 0 / 0.1);  
  
--shadow-xl: 0 20px 25px -5px rgb(0 0 0 / 0.1);
```

```
/* Transiciones */
```

```
--transition-fast: 150ms cubic-bezier(0.4, 0, 0.2, 1);
```

```
--transition-base: 250ms cubic-bezier(0.4, 0, 0.2, 1);
```

```
--transition-slow: 350ms cubic-bezier(0.4, 0, 0.2, 1);
```

```
/* Z-index scale */
```

```
--z-dropdown: 100;
```

```
--z-sticky: 200;
```

```
--z-modal: 300;
```

```
--z-popover: 400;
```

```
--z-tooltip: 500;
```

```
}
```

```
/* Base */
```

```
html {
```

```
  scroll-behavior: smooth;
```

```
  -webkit-text-size-adjust: 100%;
```

```
  text-rendering: optimizeLegibility;
```

```
  -webkit-font-smoothing: antialiased;
```

```
  -moz-osx-font-smoothing: grayscale;
```

```
}
```

```
body {
```

```
font-family: 'Inter', system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;
```

```
font-size: var(--text-base);
```

```
line-height: 1.6;
```

```
color: var(--color-text, #1f2937);
```

```
background-color: var(--color-bg, #ffffff);
```

```
}
```

```
/* Container queries para componentes responsivos */
```

```
.project-card {
```

```
  container-type: inline-size;
```

```
  container-name: project;
```

```
}
```

```
@container project (min-width: 400px) {
```

```
  .project-card__layout {
```

```
    display: grid;
```

```
    grid-template-columns: 1fr 1fr;
```

```
    gap: var(--space-md);
```

```
  }
```

```
}
```

```
@container project (min-width: 600px) {
```



```
.project-card__layout {  
  grid-template-columns: 2fr 1fr;  
}  
  
.project-card__image {  
  aspect-ratio: 16/9;  
}  
}  
  
/* Utilidades de layout */  
  
.container {  
  width: 100%;  
  max-width: var(--container-max);  
  margin-inline: auto;  
  padding-inline: var(--container-padding);  
}  
  
.stack {  
  display: flex;  
  flex-direction: column;  
  gap: var(--space, var(--space-md));  
}
```

```
.cluster {  
  
  display: flex;  
  
  flex-wrap: wrap;  
  
  gap: var(--space, var(--space-md));  
  
  align-items: center;  
  
}
```

```
.grid-auto {  
  
  display: grid;  
  
  grid-template-columns: repeat(  
    auto-fit,  
    minmax(min(var(--min-width, 300px), 100%), 1fr)  
  );  
  
  gap: var(--space, var(--space-md));  
  
}
```

/* Utilidades de texto */

```
.text-balance {  
  
  text-wrap: balance;  
  
}
```

```
.text-truncate {  
  
  overflow: hidden;
```

```
text-overflow: ellipsis;

white-space: nowrap;

}
```

```
.line-clamp-2 {

display: -webkit-box;

-webkit-line-clamp: 2;

-webkit-box-orient: vertical;

overflow: hidden;

}
```

```
/* Utilidades de visibilidad */
```

```
.sr-only {

position: absolute;

width: 1px;

height: 1px;

padding: 0;

margin: -1px;

overflow: hidden;

clip: rect(0, 0, 0, 0);

white-space: nowrap;

border-width: 0;

}
```

```
/* Reducir motion para accesibilidad */
```

```
@media (prefers-reduced-motion: reduce) {
```

```
  *,
```

```
  *::before,
```

```
  *::after {
```

```
    animation-duration: 0.01ms !important;
```

```
    animation-iteration-count: 1 !important;
```

```
    transition-duration: 0.01ms !important;
```

```
    scroll-behavior: auto !important;
```

```
  }
```

```
}
```

```
/* Modo oscuro automático si el sistema lo soporta */
```

```
@media (prefers-color-scheme: dark) {
```

```
  :root {
```

```
    --color-bg: #0f172a;
```

```
    --color-text: #f1f5f9;
```

```
  }
```

```
}
```

```
/* Utilidades de rendimiento */
```

```
.gpu-accelerated {
```

```
transform: translateZ(0);  
backface-visibility: hidden;  
will-change: transform;  
}
```

```
.content-visibility-auto {  
    content-visibility: auto;  
    contain-intrinsic-size: 0 500px;  
}
```

/ Optimización de imágenes */*

```
img {  
    max-width: 100%;  
    height: auto;  
    vertical-align: middle;  
    font-style: italic;  
    background-repeat: no-repeat;  
    background-size: cover;  
    shape-margin: 1rem;  
}
```

/ Focus visible para accesibilidad */*

```
:focus:not(:focus-visible) {
```

```
outline: none;  
}
```

```
:focus-visible {  
  outline: 2px solid var(--color-primary);  
  outline-offset: 2px;  
}
```

```
/* Touch targets mínimos */
```

```
button,  
a,  
input,  
textarea,  
select {  
  min-height: 44px;  
  min-width: 44px;  
}
```

```
@media (pointer: fine) {  
  button,  
  a,  
  input,  
  textarea,
```

```
select {  
    min-height: auto;  
    min-width: auto;  
}  
}  
...
```

5.2 Animaciones Optimizadas (solo GPU)

```
```css  

/* styles/animations.css - Cargado asíncronamente */

/* Fade in up - usada en hero */
@keyframes fadeInUp {
 from {
 opacity: 0;
 transform: translateY(30px);
 }
 to {
 opacity: 1;
 transform: translateY(0);
 }
}
```

```
.animate-fade-in-up {
 animation: fadeInUp 0.6s var(--transition-base) forwards;
 will-change: opacity, transform;
}
```

```
/* Stagger children */
```

```
.stagger-children > * {
 opacity: 0;
 animation: fadeInUp 0.5s var(--transition-base) forwards;
}
```

```
.stagger-children > *:nth-child(1) { animation-delay: 0.1s; }
.stagger-children > *:nth-child(2) { animation-delay: 0.2s; }
.stagger-children > *:nth-child(3) { animation-delay: 0.3s; }
.stagger-children > *:nth-child(4) { animation-delay: 0.4s; }
```

```
/* Hover lift effect */
```

```
.hover-lift {
 transition: transform var(--transition-base), box-shadow var(--transition-base);
 will-change: transform;
}
```



```
.hover-lift:hover {
 transform: translateY(-4px);
 box-shadow: var(--shadow-xl);
}
```

```
/* Skeleton loading shimmer */
```

```
@keyframes shimmer {
 0% {
 background-position: -200% 0;
 }
 100% {
 background-position: 200% 0;
 }
}
```

```
.skeleton {
 background: linear-gradient(
 90deg,
 #f0f0f0 25%,
 #e0e0e0 50%,
 #f0f0f0 75%
);
 background-size: 200% 100%;
```

```
 animation: shimmer 1.5s infinite;
}

/* Scroll reveal - Intersection Observer en CSS */

.reveal {

 opacity: 0;

 transform: translateY(20px);

 transition: opacity 0.6s var(--transition-base),
 transform 0.6s var(--transition-base);
}

.reveal.is-visible {

 opacity: 1;

 transform: translateY(0);
}

/* Parallax suave */

.parallax {

 transform: translateZ(0);

 will-change: transform;
}

/* Carousel optimizado */
```

```
.carousel-track {
 display: flex;
 gap: var(--space-md);
 overflow-x: auto;
 scroll-snap-type: x mandatory;
 -webkit-overflow-scrolling: touch;
 scrollbar-width: none;
 -ms-overflow-style: none;
}
```

```
.carousel-track::-webkit-scrollbar {
 display: none;
}
```

```
.carousel-item {
 flex: 0 0 auto;
 scroll-snap-align: start;
 width: min(85vw, 400px);
}
```

/\* Progress bar animation \*/

```
@keyframes progress {
 from { transform: scaleX(0); }
```

```
to { transform: scaleX(1); }
}
```

```
.progress-bar {
 transform-origin: left;
 animation: progress 2s var(--transition-base) forwards;
}
...

```

## ## 🔒 PARTE 6: SERVICE WORKER Y CACHE

### ### 6.1 Service Worker Estratégico Avanzado

```
````javascript  
// public/sw.js  
  
const CACHE_VERSION = 'transervica-v2';  
  
const STATIC_CACHE = `${CACHE_VERSION}-static`;  
  
const IMAGE_CACHE = `${CACHE_VERSION}-images`;  
  
const API_CACHE = `${CACHE_VERSION}-api`;  
  
  
// Estrategias de caché
```

```
const STRATEGIES = {  
  
  // Cache First para assets estáticos  
  
  CACHE_FIRST: async (request, cacheName) => {  
  
    const cache = await caches.open(cacheName);  
  
    const cached = await cache.match(request);  
  
  
    if (cached) {  
  
      // Revalidar en background  
  
      fetch(request).then(response => {  
  
        if (response.ok) cache.put(request, response);  
  
      });  
  
      return cached;  
    }  
  
  
    const response = await fetch(request);  
  
    if (response.ok) {  
  
      cache.put(request, response.clone());  
  
    }  
  
    return response;  
  },  
  
  
  // Network First para APIs  
  
  NETWORK_FIRST: async (request, cacheName, maxAge = 3600) => {
```

```
try {  
  
  const networkResponse = await fetch(request);  
  
  if (networkResponse.ok) {  
  
    const cache = await caches.open(cacheName);  
  
    const cloned = networkResponse.clone();  
  
    cloned.headers.set('sw-cached-at', Date.now().toString());  
  
    cache.put(request, cloned);  
  
  }  
  
  return networkResponse;  
} catch (error) {  
  
  const cache = await caches.open(cacheName);  
  
  const cached = await cache.match(request);  
  
  
  if (cached) {  
  
    const cachedAt = parseInt(cached.headers.get('sw-cached-at') || '0');  
  
    const age = (Date.now() - cachedAt) / 1000;  
  
  
    if (age < maxAge) {  
  
      return cached;  
  
    }  
  
  }  
  
  throw error;  
}
```

```
},
```

```
// Stale While Revalidate para imágenes
```

```
STALE_WHILE_REVALIDATE: async (request, cacheName) => {
```

```
  const cache = await caches.open(cacheName);
```

```
  const cached = await cache.match(request);
```

```
  const networkPromise = fetch(request).then(response => {
```

```
    if (response.ok) {
```

```
      cache.put(request, response.clone());
```

```
    }
```

```
    return response;
```

```
  }).catch(() => cached);
```

```
  return cached || networkPromise;
```

```
}
```

```
};
```

```
// Instalación - Precache crítico
```

```
self.addEventListener('install', (event) => {
```

```
  event.waitUntil(
```

```
    caches.open(STATIC_CACHE).then(cache => {
```

```
      return cache.addAll([
```

```

    '/',

    '/index.html',

    '/assets/critical.css',

    '/assets/logo-transervica-200.webp',

    '/assets/logo-transervica-128.webp',

    '/offline.html'

  ]);

  }).then(() => self.skipWaiting())

);

});

// Activación - Limpiar caches antiguas
self.addEventListener('activate', (event) => {
  event.waitUntil(
    caches.keys().then(cacheNames => {
      return Promise.all(
        cacheNames
          .filter(name => name.startsWith('transervica-') && !name.includes(CACHE_VERSION))
          .map(name => caches.delete(name))
      );
    })
  ).then(() => self.clients.claim())
);
});

```



```
// Fetch - Estrategias por tipo de recurso

self.addEventListener('fetch', (event) => {

  const { request } = event;

  const url = new URL(request.url);

  // Ignorar requests no GET

  if (request.method !== 'GET') return;

  // Estrategia para HTML (Network First con fallback offline)

  if (request.mode === 'navigate') {

    event.respondWith(

      fetch(request)

        .catch(() => caches.match('/offline.html'))

    );

    return;

  }

  // Assets estáticos (JS, CSS)

  if (url.pathname.match(/\.(js|css)$/)) {

    event.respondWith(STRATEGIES.CACHE_FIRST(request, STATIC_CACHE));

    return;

  }

}
```

```
// Imágenes (Stale While Revalidate)
```

```
if (request.destination === 'image') {
```

```
  event.respondWith(STRATEGIES.STALE_WHILE_REVALIDATE(request, IMAGE_CACHE));
```

```
  return;
```

```
}
```

```
// APIs (Network First con cache 1 hora)
```

```
if (url.pathname.startsWith('/api/')) {
```

```
  event.respondWith(STRATEGIES.NETWORK_FIRST(request, API_CACHE, 3600));
```

```
  return;
```

```
}
```

```
// Fonts (Cache First)
```

```
if (request.destination === 'font') {
```

```
  event.respondWith(STRATEGIES.CACHE_FIRST(request, STATIC_CACHE));
```

```
  return;
```

```
}
```

```
});
```

```
// Background Sync para formularios offline
```

```
self.addEventListener('sync', (event) => {
```

```
  if (event.tag === 'sync-forms') {
```

```
    event.waitUntil(syncFormSubmissions());  
  }  
});
```

```
async function syncFormSubmissions() {  
  const db = await openDB('form-queue', 1);  
  const submissions = await db.getAll('submissions');  
  
  for (const submission of submissions) {  
    try {  
      await fetch('/api/contact', {  
        method: 'POST',  
        headers: { 'Content-Type': 'application/json' },  
        body: JSON.stringify(submission.data)  
      });  
      await db.delete('submissions', submission.id);  
    } catch (error) {  
      console.error('Sync failed for submission:', submission.id);  
    }  
  }  
}
```

```
// Push notifications (opcional)
```

```

self.addEventListener('push', (event) => {

  const data = event.data.json();

  event.waitUntil(

    self.registration.showNotification(data.title, {

      body: data.body,

      icon: '/assets/logo-192.png',

      badge: '/assets/badge-72.png',

      data: data.url

    })

  );

});

```

```

self.addEventListener('notificationclick', (event) => {

  event.notification.close();

  event.waitUntil(

    clients.openWindow(event.notification.data)

  );

});

...

```

6.2 IndexedDB para Datos Offline

```
```javascript
```

```
// utils/offlineStorage.js

import { openDB } from 'idb';

const DB_NAME = 'transervica-offline';

const DB_VERSION = 1;

export const initDB = async () => {

 return openDB(DB_NAME, DB_VERSION, {

 upgrade(db) {

 // Store para proyectos

 if (!db.objectStoreNames.contains('projects')) {

 const projectStore = db.createObjectStore('projects', { keyPath: 'id' });

 projectStore.createIndex('by-date', 'date', { unique: false });

 }

 // Store para formularios pendientes

 if (!db.objectStoreNames.contains('pending-forms')) {

 db.createObjectStore('pending-forms', {

 keyPath: 'id',

 autoIncrement: true

 });

 }

 }

 });

}
```

```
// Store para caché de imágenes

if (!db.objectStoreNames.contains('image-cache')) {

 db.createObjectStore('image-cache', { keyPath: 'url' });

}

});

};
```

```
export const cacheProject = async (project) => {

 const db = await initDB();

 await db.put('projects', {

 ...project,

 cachedAt: Date.now()

 });

};
```

```
export const getCachedProjects = async () => {

 const db = await initDB();

 return db.getAll('projects');

};
```

```
export const queueFormSubmission = async (formData) => {

 const db = await initDB();
```

```
await db.add('pending-forms', {
 data: formData,
 timestamp: Date.now(),
 retries: 0
});
```

```
// Registrar para sync cuando haya conexión
if ('serviceWorker' in navigator && 'SyncManager' in window) {
 const registration = await navigator.serviceWorker.ready;
 await registration.sync.register('sync-forms');
}
};
...
```

---

## ## 📊 PARTE 7: MONITOREO Y MÉTRICAS

### ### 7.1 Sistema de Analytics de Rendimiento

```
```javascript
```

```
// utils/performanceMonitor.js
```

```
class PerformanceMonitor {
```

```
constructor() {  
  
  this.metrics = {};  
  
  this.observers = [];  
  
  this.init();  
  
}
```

```
init() {  
  
  // Web Vitals  
  
  this.observeLCP();  
  
  this.observeFID();  
  
  this.observeCLS();  
  
  this.observeINP();  
  
  this.observeTTFB();  
  
  
  // Métricas personalizadas  
  
  this.observeCustomMetrics();  
  
  
  // Enviar métricas cuando la página se oculta  
  
  document.addEventListener('visibilitychange', () => {  
  
    if (document.visibilityState === 'hidden') {  
  
      this.sendMetrics();  
  
    }  
  
  });  
}
```



```
}
```

```
observeLCP() {
```

```
  if (!('PerformanceObserver' in window)) return;
```

```
  const observer = new PerformanceObserver((list) => {
```

```
    const entries = list.getEntries();
```

```
    const lastEntry = entries[entries.length - 1];
```

```
    this.metrics.lcp = {
```

```
      value: lastEntry.startTime,
```

```
      element: lastEntry.element?.tagName,
```

```
      url: lastEntry.url,
```

```
      timestamp: Date.now()
```

```
    };
```

```
    // Marcar en el DOM para debugging
```

```
    if (lastEntry.element) {
```

```
      lastEntry.element.setAttribute('data-lcp', 'true');
```

```
    }
```

```
  });
```

```
  observer.observe({ entryTypes: ['largest-contentful-paint'] });
```

```
    this.observers.push(observer);  
  }
```

```
observeFID() {  
  if (!('PerformanceObserver' in window)) return;  
  
  const observer = new PerformanceObserver((list) => {  
    for (const entry of list.getEntries()) {  
      if (entry.entryType === 'first-input') {  
        this.metrics.fid = {  
          value: entry.processingStart - entry.startTime,  
          type: entry.name,  
          timestamp: Date.now()  
        };  
      }  
    }  
  });  
};
```

```
observer.observe({ entryTypes: ['first-input'] });  
this.observers.push(observer);  
}
```

```
observeCLS() {
```

```
if (!('PerformanceObserver' in window)) return;
```

```
let clsValue = 0;
```

```
let clsEntries = [];
```

```
const observer = new PerformanceObserver((list) => {
```

```
  for (const entry of list.getEntries()) {
```

```
    if (!entry.hadRecentInput) {
```

```
      clsValue += entry.value;
```

```
      clsEntries.push({
```

```
        value: entry.value,
```

```
        sources: entry.sources.map(s => ({
```

```
          node: s.node?.tagName,
```

```
          previousRect: s.previousRect,
```

```
          currentRect: s.currentRect
```

```
        })))
```

```
    });
```

```
  }
```

```
}
```

```
this.metrics.cls = {
```

```
  value: clsValue,
```

```
  entries: clsEntries,
```

```
    timestamp: Date.now()

  };

});
```

```
observer.observe({ entryTypes: ['layout-shift'] });

this.observers.push(observer);
}
```

```
observeINP() {

  if (!('PerformanceObserver' in window)) return;

  let interactions = [];

  const observer = new PerformanceObserver((list) => {

    for (const entry of list.getEntries()) {

      if (entry.entryType === 'event' && entry.interactionId) {

        interactions.push({

          id: entry.interactionId,

          duration: entry.processingEnd - entry.startTime,

          type: entry.name

        });

      }

    }

  })
}
```

```

// Calcula INP (Interaction to Next Paint)

if (interactions.length > 0) {

  const sorted = interactions.sort((a, b) => b.duration - a.duration);

  this.metrics.inp = {

    value: sorted[Math.floor(sorted.length * 0.98)]?.duration || 0,

    interactions: interactions.length,

    timestamp: Date.now()

  };

}

});

observer.observe({

  entryTypes: ['event'],

  buffered: true,

  durationThreshold: 0

});

this.observers.push(observer);

}

observeTTFB() {

  const navigation = performance.getEntriesByType('navigation')[0];

  if (navigation) {

```

```

this.metrics.ttfb = {
  value: navigation.responseStart - navigation.startTime,
  dns: navigation.domainLookupEnd - navigation.domainLookupStart,
  tcp: navigation.connectEnd - navigation.connectStart,
  ssl: navigation.secureConnectionStart > 0
    ? navigation.connectEnd - navigation.secureConnectionStart
    : 0,
  timestamp: Date.now()
};
}
}

```

```

observeCustomMetrics() {
  // Tiempo hasta que el hero es visible
  const heroObserver = new IntersectionObserver((entries) => {
    entries.forEach(entry => {
      if (entry.isIntersecting && !this.metrics.heroVisible) {
        this.metrics.heroVisible = {
          value: performance.now(),
          timestamp: Date.now()
        };
        heroObserver.disconnect();
      }
    });
  });
}

```

```
});
```

```
});
```

```
const hero = document.querySelector('.hero');
```

```
if (hero) heroObserver.observe(hero);
```

```
// Tiempo hasta interactivo
```

```
if ('requestIdleCallback' in window) {
```

```
  requestIdleCallback(() => {
```

```
    this.metrics.timeToInteractive = {
```

```
      value: performance.now(),
```

```
      timestamp: Date.now()
```

```
    };
```

```
  });
```

```
}
```

```
// Errores de carga de recursos
```

```
window.addEventListener('error', (e) => {
```

```
  if (!this.metrics.resourceErrors) this.metrics.resourceErrors = [];
```

```
  this.metrics.resourceErrors.push({
```

```
    message: e.message,
```

```
    filename: e.filename,
```

```
    timestamp: Date.now()
```

```
});
```

```
}, true);
```

```
}
```

```
sendMetrics() {
```

```
    // Enviar a analytics solo si hay métricas significativas
```

```
    if (Object.keys(this.metrics).length === 0) return;
```

```
    const payload = {
```

```
        url: window.location.href,
```

```
        userAgent: navigator.userAgent,
```

```
        connection: navigator.connection ? {
```

```
            effectiveType: navigator.connection.effectiveType,
```

```
            saveData: navigator.connection.saveData
```

```
        } : null,
```

```
        metrics: this.metrics,
```

```
        sessionId: this.getSessionId()
```

```
    };
```

```
    // Usar sendBeacon para envío confiable
```

```
    if (navigator.sendBeacon) {
```

```
        navigator.sendBeacon('/api/analytics/performance', JSON.stringify(payload));
```

```
    } else {
```



```
fetch('/api/analytics/performance', {  
  method: 'POST',  
  body: JSON.stringify(payload),  
  keepalive: true,  
  headers: { 'Content-Type': 'application/json' }  
}).catch(() => {});  
}
```

```
// También enviar a console en desarrollo  
if (process.env.NODE_ENV === 'development') {  
  console.table(this.metrics);  
}  
}
```

```
getSessionId() {  
  let id = sessionStorage.getItem('perf-session-id');  
  if (!id) {  
    id = Math.random().toString(36).substr(2, 9);  
    sessionStorage.setItem('perf-session-id', id);  
  }  
  return id;  
}
```

```
destroy() {  
  this.observers.forEach(obs => obs.disconnect());  
}  
}
```

// Instancia global

```
export const perfMonitor = new PerformanceMonitor();
```

// Hook para React

```
export function usePerformanceMetric(name) {  
  useEffect(() => {  
    const start = performance.now();  
  
    return () => {  
      const duration = performance.now() - start;  
  
      perfMonitor.metrics[`component-${name}`] = {  
        value: duration,  
        timestamp: Date.now()  
      };  
    };  
  }, [name]);  
}
```

...

7.2 Dashboard de Monitoreo

```
````javascript

// components/PerformanceDashboard.jsx (solo desarrollo)

export function PerformanceDashboard() {

 const [metrics, setMetrics] = useState({});

 useEffect(() => {

 const interval = setInterval(() => {

 setMetrics({ ...perfMonitor.metrics });

 }, 1000);

 return () => clearInterval(interval);

 }, []);

 const getRating = (metric, value) => {

 const thresholds = {

 lcp: { good: 2500, poor: 4000 },

 fid: { good: 100, poor: 300 },

 cls: { good: 0.1, poor: 0.25 },

 inp: { good: 200, poor: 500 },

 ttfb: { good: 600, poor: 1000 }

 };

 }
```

```
const t = thresholds[metric];

if (!t) return 'neutral';

if (value <= t.good) return 'good';

if (value <= t.poor) return 'needs-improvement';

return 'poor';
};
```

```
const formatValue = (metric, value) => {

 if (metric === 'cls') return value.toFixed(3);

 if (metric === 'lcp' || metric === 'fid' || metric === 'inp' || metric === 'ttfb') {

 return `${Math.round(value)}ms`;

 }

 return value;
};
```

```
return (

 <div style={{

 position: 'fixed',

 bottom: '20px',

 right: '20px',

 background: 'rgba(0,0,0,0.9)',

 color: 'white',
```

```
padding: '16px',

borderRadius: '8px',

fontFamily: 'monospace',

fontSize: '12px',

zIndex: 9999,

maxWidth: '300px'

}}>

<h3 style={{ margin: '0 0 12px 0', fontSize: '14px' }}>Core Web Vitals</h3>

{Object.entries(metrics).map(([key, data]) => {

 if (!data?.value) return null;

 const rating = getRating(key, data.value);

 const colors = {

 good: '#22c55e',

 'needs-improvement': '#eab308',

 poor: '#ef4444',

 neutral: '#6b7280'

 };

 return (

 <div key={key} style={{

 display: 'flex',

 justifyContent: 'space-between',

 marginBottom: '8px',
```

```

 alignItems: 'center'

 }}>

 {key}

 <span style={{

 color: colors[rating],

 fontWeight: 'bold'

 }}>

 {formatValue(key, data.value)}

</div>

);

}}

</div>

);

}

...

```

## ## 🚧 PARTE 8: TESTING Y VALIDACIÓN

### ### 8.1 Suite de Tests de Rendimiento

```
````javascript

// tests/performance.spec.js

import { test, expect } from '@playwright/test';

test.describe('Performance Tests', () => {

  test('LCP should be under 2.5s', async ({ page }) => {

    await page.goto('/');

    const lcp = await page.evaluate(() => {

      return new Promise((resolve) => {

        new PerformanceObserver((list) => {

          const entries = list.getEntries();

          resolve(entries[entries.length - 1].startTime);

        }).observe({ entryTypes: ['largest-contentful-paint'] });

      });

    });

    expect(lcp).toBeLessThan(2500);

  });

  test('No layout shifts after initial load', async ({ page }) => {

    await page.goto('/');

    await page.waitForLoadState('networkidle');
```

```

const cls = await page.evaluate(() => {

  let clsValue = 0;

  new PerformanceObserver((list) => {

    for (const entry of list.getEntries()) {

      if (!entry.hadRecentInput) clsValue += entry.value;

    }

  }).observe({ entryTypes: ['layout-shift'] });

  // Esperar 5 segundos

  return new Promise(resolve => setTimeout(() => resolve(clsValue), 5000));

});

expect(cls).toBeLessThan(0.1);

});

test('JavaScript bundle size under 200KB', async ({ page }) => {

  const responses = [];

  page.on('response', response => {

    if (response.url().includes('.js')) {

      responses.push(response);

    }

  });

});

```



```
await page.goto('/');

await page.waitForLoadState('networkidle');


for (const response of responses) {

  const body = await response.body();

  expect(body.length).toBeLessThan(200 * 1024); // 200KB
}

});


test('Images should have proper sizing', async ({ page }) => {

  await page.goto('/');


  const images = await page.$$eval('img', imgs =>

    imgs.map(img => ({

      src: img.src,

      naturalWidth: img.naturalWidth,

      naturalHeight: img.naturalHeight,

      clientWidth: img.clientWidth,

      clientHeight: img.clientHeight,

      loading: img.loading,

      decoding: img.decoding

    })))

});
```

```
);
```

```
for (const img of images) {
```

```
  // Verificar que no se cargan imágenes más grandes de lo necesario
```

```
  const oversizeRatio = img.naturalWidth / img.clientWidth;
```

```
  expect(oversizeRatio).toBeLessThan(2); // Máximo 2x para retina
```

```
  // Verificar lazy loading en imágenes below-fold
```

```
  if (!img.src.includes('logo') && !img.src.includes('hero')) {
```

```
    expect(img.loading).toBe('lazy');
```

```
  }
```

```
}
```

```
});
```

```
test('Service Worker registration', async ({ page }) => {
```

```
  const swRegistered = await page.evaluate(() => {
```

```
    return navigator.serviceWorker.ready.then(() => true);
```

```
  });
```

```
  expect(swRegistered).toBe(true);
```

```
});
```

```
});
```

```
...
```

8.2 Lighthouse CI

```
```javascript
// lighthouserc.js

module.exports = {

 ci: {

 collect: {

 url: [

 'http://localhost:3000/',

 'http://localhost:3000/proyectos',

 'http://localhost:3000/contacto'

],

 numberOfRuns: 3,

 settings: {

 preset: 'desktop',

 throttling: {

 cpuSlowdownMultiplier: 1,

 downloadThroughputKbps: 5000,

 uploadThroughputKbps: 1000

 },

 screenEmulation: {

 mobile: false,
```

```
width: 1350,

height: 940,

deviceScaleFactor: 1,

disabled: false

},

formFactor: 'desktop'

}

},

assert: {

 assertions: {

 'categories:performance': ['error', { minScore: 0.95 }],

 'categories:accessibility': ['error', { minScore: 0.95 }],

 'categories:best-practices': ['error', { minScore: 0.95 }],

 'categories:seo': ['error', { minScore: 0.95 }],

 'first-contentful-paint': ['error', { maxNumericValue: 1800 }],

 'largest-contentful-paint': ['error', { maxNumericValue: 2500 }],

 'cumulative-layout-shift': ['error', { maxNumericValue: 0.1 }],

 'total-blocking-time': ['error', { maxNumericValue: 200 }],

 'speed-index': ['error', { maxNumericValue: 3400 }],

 'unused-javascript': 'warn',

 'unused-css-rules': 'warn',

 'modern-image-formats': 'error',

 'uses-text-compression': 'error',
```

```
'uses-responsive-images': 'error',
'efficiently-encodes-images': 'error',
'uses-optimized-images': 'error',
'uses-long-cache-ttl': 'warn',
'total-byte-weight': ['error', { maxNumericValue: 1000000 }] // 1MB
}
},
upload: {
 target: 'temporary-public-storage',
 githubAppToken: process.env.LHCI_GITHUB_APP_TOKEN
}
}
};
...

```

## ## 📋 PARTE 9: CHECKLIST DE IMPLEMENTACIÓN

### ### Fase 1: Fundamentos (Días 1-3)

- [ ] Implementar HTML crítico inline (< 14KB)
- [ ] Optimizar logo en 5 tamaños (128, 160, 200, 256, 400px)
- [ ] Configurar preconnect y dns-prefetch

- [ ] Implementar fuentes con font-display: swap
- [ ] Crear Service Worker básico

#### ### Fase 2: Assets (Días 4-6)

- [ ] Pipeline de optimización de imágenes (WebP, AVIF)
- [ ] Implementar componente OptimizedImage con srcset
- [ ] Generar placeholders LQIP para todas las imágenes
- [ ] Reemplazar iframes de YouTube con LiteYouTube
- [ ] Optimizar videos (posters, lazy load)

#### ### Fase 3: JavaScript (Días 7-9)

- [ ] Configurar code splitting en Vite/Webpack
- [ ] Implementar lazy loading de rutas
- [ ] Mover lógica pesada a Web Workers
- [ ] Eliminar código muerto (tree shaking)
- [ ] Implementar prefetching inteligente

#### ### Fase 4: Polish (Días 10-12)

- [ ] Implementar CSS crítico y async
- [ ] Optimizar animaciones (GPU only)
- [ ] Fix accesibilidad (contraste, aria-labels, focus)
- [ ] Implementar modo oscuro (opcional)
- [ ] Testing en dispositivos reales

### ### Fase 5: Monitoreo (Día 13-14)

- [ ] Implementar Web Vitals monitoring
- [ ] Configurar Lighthouse CI
- [ ] Tests automatizados de rendimiento
- [ ] Documentación para el equipo

---

### ## 🎯 RESULTADOS ESPERADOS DETALLADOS

#### ### Antes vs Después

Aspecto	Antes	Después	Impacto
-----	-----	-----	-----
**Puntuación Performance**	56/100	98-100/100	+42-44 puntos
**LCP (Móvil)**	6.6s	1.2-1.8s	-73%
**FCP (Móvil)**	2.8s	0.8-1.2s	-57%
**TBT (Móvil)**	440ms	80-120ms	-73%
**CLS**	0	0	Perfecto
**Tamaño Total**	3.1MB	450-600KB	-80%
**Peticiónes HTTP**	45	15-20	-56%
**Tiempo 3G**	~15s	~2.5s	-83%


| \*\*Time to Interactive\*\* | 8.5s | 1.5-2s | -76% |

### ### Business Impact Estimado

- \*\*Bounce Rate:\*\* Reducción del 32% (mejora de 1s = -7% bounce rate)
- \*\*Conversion Rate:\*\* Incremento del 15-20%
- \*\*SEO Ranking:\*\* Mejora significativa en Core Web Vitals
- \*\*Costos de Hosting:\*\* Reducción del 70% en transferencia de datos

---

### Obras citadas

1. Solving 404 Errors on Refresh in React Vite Apps | by Viranga Jayawardana - Medium, fecha de acceso: febrero 9, 2026, <https://virangaj.medium.com/solving-404-errors-on-refresh-in-react-vite-apps-c52fc596dc27>
2. Deploy Vite React project with React router without 404 on Github pages - DEV Community, fecha de acceso: febrero 9, 2026, <https://dev.to/avithe1/deploy-vite-react-project-with-react-router-without-404-on-github-pages-h6p>
3. SPA Routing 404s: Taming the "Page Not Found" Monster with `\_redirects` & Server Magic  - DEV Community, fecha de acceso: febrero 9, 2026, [https://dev.to/alex\\_aslam/spa-routing-404s-taming-the-page-not-found-monster-with-redirects-server-magic-c01](https://dev.to/alex_aslam/spa-routing-404s-taming-the-page-not-found-monster-with-redirects-server-magic-c01)
4. How to fix Crawled - currently not indexed – ZipTie.dev, fecha de acceso: febrero 9, 2026, <https://ziptie.dev/blog/how-to-fix-crawled-currently-not-indexed/>
5. fecha de acceso: febrero 9, 2026, <https://delimp.com/blog/how-to-fix-the-crawled-currently-not-indexed-issue-in-gsc-5-easy-steps/>
6. Static Deployment Configuration - Replit Docs, fecha de acceso: febrero 9, 2026, <https://docs.replit.com/cloud-services/deployments/static-deployments-advanced>
7. How We Resolve 'Alternative Page With Proper Canonical Tag' on Google Search Console, fecha de acceso: febrero 9, 2026, <https://www.embarque.io/post/resolve-alternative-page-with-proper-canonical-tag-on-google-search-console>



8. How to fix the 'Alternate page with proper canonical tag' error in Google Search Console, fecha de acceso: febrero 9, 2026,  
<https://searchengineland.com/how-to-fix-the-alternate-page-with-proper-canonical-tag-error-in-google-search-console-448852>
9. 9 Non-Obvious Fixes for "Crawled / Discovered - Currently Not Indexed" - Motava, fecha de acceso: febrero 9, 2026,  
<https://www.motava.com/blog/fixes-discovered-currently-not-indexed-urls/>
10. react-lite-youtube-embed - NPM, fecha de acceso: febrero 9, 2026,  
<https://www.npmjs.com/package/react-lite-youtube-embed?activeTab=dependencies>
11. YouTube Embeds are Bananas Heavy and it's Fixable - Frontend Masters Blog, fecha de acceso: febrero 9, 2026,  
<https://frontendmasters.com/blog/youtube-embeds-are-bananas-heavy-and-its-fixable/>
12. ibrahimcesar/react-lite-youtube-embed - GitHub, fecha de acceso: febrero 9, 2026, <https://github.com/ibrahimcesar/react-lite-youtube-embed>
13. Optimizing YouTube embed performance in Sanity with React Portable Text, fecha de acceso: febrero 9, 2026,  
<https://www.sanity.io/answers/how-to-improve-performance-when-rendering-youtube-videos-in-nextjs-with-sanity>
14. Displays images with incorrect aspect ratio | Lighthouse - Chrome for Developers, fecha de acceso: febrero 9, 2026,  
<https://developer.chrome.com/docs/lighthouse/best-practices/image-aspect-ratio>
15. Image elements do not have explicit width and height · tailwindlabs/tailwindcss · Discussion #4658 - GitHub, fecha de acceso: febrero 9, 2026,  
<https://github.com/tailwindlabs/tailwindcss/discussions/4658>
16. How To Optimize Images in Vite Projects? - GeeksforGeeks, fecha de acceso: febrero 9, 2026,  
<https://www.geeksforgeeks.org/javascript/how-to-optimize-images-in-vite-projects/>
17. SEO Optimization for React + Vite Apps - DEV Community, fecha de acceso: febrero 9, 2026,  
[https://dev.to/ali\\_dz/optimizing-seo-in-a-react-vite-project-the-ultimate-guide-3mbh](https://dev.to/ali_dz/optimizing-seo-in-a-react-vite-project-the-ultimate-guide-3mbh)
18. vite-plugin-sitemap - npm, fecha de acceso: febrero 9, 2026,  
<https://www.npmjs.com/package/vite-plugin-sitemap>
19. SEO canonical tags not appearing in DOM in Lovable (React + Vite SPA) - react-helmet-async issue? : r/seogrowth - Reddit, fecha de acceso: febrero 9, 2026,  
[https://www.reddit.com/r/seogrowth/comments/1qrzbcw/seo\\_canonical\\_tags\\_not\\_appearing\\_in\\_dom\\_in/](https://www.reddit.com/r/seogrowth/comments/1qrzbcw/seo_canonical_tags_not_appearing_in_dom_in/)
20. Web Accessibility Color Contrast Checker - Conform to WCAG, fecha de acceso: febrero 9, 2026, <https://accessibleweb.com/color-contrast-checker/>
21. Check Text and Background for Sufficient Color Contrast | Accessibility Tips -

- Deque University, fecha de acceso: febrero 9, 2026,  
<https://dequeuniversity.com/tips/color-contrast>
22. Contrast Checker - WebAIM, fecha de acceso: febrero 9, 2026,  
<https://webaim.org/resources/contrastchecker/>
  23. Accessibility and Color Contrast Guidelines - The University of Maryland, Baltimore, fecha de acceso: febrero 9, 2026,  
<https://www.umaryland.edu/cpa/branding-hub/colors-and-design/accessibility-and-color-contrast-guidelines/>
  24. Learn Making SwiperJS Accessible | Interactivity, Accessibility, and ..., fecha de acceso: febrero 9, 2026,  
<https://codefinity.com/courses/v2/f4b09f6a-1d4e-4891-820c-4a9a0104bd47/1b804062-ecd4-4fbb-8bb9-02524bdc8f6f/a3e8afb1-8d53-4c91-b202-f7ce7827769f>
  25. How does one fix "policy violation: accelerometer is not allowed in this document." (vanilla-tilt.min) - Stack Overflow, fecha de acceso: febrero 9, 2026,  
<https://stackoverflow.com/questions/74827837/how-does-one-fix-policy-violation-accelerometer-is-not-allowed-in-this-documen>
  26. Control browser features with Permissions Policy | Privacy & Security, fecha de acceso: febrero 9, 2026,  
<https://developer.chrome.com/docs/privacy-security/permissions-policy>
  27. How to remove the "Made in Replit" popup - Reddit, fecha de acceso: febrero 9, 2026,  
[https://www.reddit.com/r/replit/comments/1ju9j2l/how\\_to\\_remove\\_the\\_made\\_in\\_replit\\_popup/](https://www.reddit.com/r/replit/comments/1ju9j2l/how_to_remove_the_made_in_replit_popup/)
  28. How to Remove the "Made with Replit" Watermark - YouTube, fecha de acceso: febrero 9, 2026, <https://www.youtube.com/watch?v=RSvW9liU2cl>
  29. I found how to keep dev features out of production : r/replit - Reddit, fecha de acceso: febrero 9, 2026,  
[https://www.reddit.com/r/replit/comments/1j6r32r/i\\_found\\_how\\_to\\_keep\\_dev\\_features\\_out\\_of\\_production/](https://www.reddit.com/r/replit/comments/1j6r32r/i_found_how_to_keep_dev_features_out_of_production/)

## # Informe de Optimización de Rendimiento para grupotranservica.com

\*\*Objetivo:\*\* Llevar la puntuación de rendimiento al 100% en móvil y escritorio, priorizando una carga ultra-rápida y una experiencia fluida en todos los dispositivos móviles.

## ## Resumen Ejecutivo

- \*\*Puntuación Actual (Móvil):\*\* 56 (Rendimiento), 88 (Accesibilidad), 69 (Prácticas recomendadas), 100 (SEO).

- \*\*Puntuación Actual (Escritorio):\*\* 71 (Rendimiento), 88 (Accesibilidad), 69 (Prácticas recomendadas), 100 (SEO).

- \*\*Problemas Críticos:\*\* Imágenes sobredimensionadas, exceso de JavaScript (principalmente de YouTube), solicitudes que bloquean el renderizado y código no utilizado.

---

### ## 1. Optimización de Imágenes (Impacto: Alto)

\*\*Problema:\*\* Las imágenes son demasiado grandes para las dimensiones mostradas, retrasando el LCP y aumentando el peso de la página.

- \*\*Acciones Inmediatas:\*\*

1. \*\*Redimensionar y Comprimir:\*\*

- `2\_1754167186024-CI4L8huF.webp` (261 KiB): Reducir de 1920x1081 a 665x499. Ahorro estimado: ~220 KiB.

- `logo%20transervica%20sin%20fondo\_1754163034585-PZeMbfCv.webp` (105 KiB): Reducir de 1738x1482 a 197x168. Ahorro estimado: ~104 KiB.

- Usar herramientas como Squoosh.app o Sharp para generar versiones WebP optimizadas.

2. \*\*Imágenes Adaptables (Responsive Images):\*\*

- Implementar `srcset` con múltiples resoluciones. Ejemplo:

```html



```

3. \*\*Prioridad Crítica:\*\*

- Para el logo LCP, asegurar `fetchpriority="high"` y evitar `loading="lazy"` en la imagen visible inicial.

---

### ## 2. Reducción de JavaScript (Impacto: Muy Alto)

\*\*Problema:\*\* YouTube y scripts propios consumen >1.5 MB y bloquean el hilo principal (TBT > 400 ms).

- \*\*Acciones Inmediatas:\*\*

1. \*\*Carga Diferida de YouTube:\*\*

- Reemplazar iframes de YouTube por una miniatura con `loading="lazy"`. Al hacer clic, cargar el iframe dinámicamente. Ejemplo:

```html

<div class="youtube-placeholder" data-video-id="VIDEO_ID">

</div>

<script>

document.querySelector('.youtube-placeholder').addEventListener('click',
function() {

 this.innerHTML = `<iframe

src="https://www.youtube.com/embed/VIDEO_ID?autoplay=1" ...></iframe>`;

 });

</script>

```

- Esto elimina ~1.4 MB de JS y reduce el TBT drásticamente.

2. \*\*Eliminar JavaScript No Utilizado:\*\*

- Usar herramientas como `coverage` en DevTools para identificar código no ejecutado.

- Aplicar tree-shaking y code splitting en el build. Ejemplo con Vite/Webpack:

```
...js
import('./module.js').then(module => module.init());
...
```

### 3. **\*\*Minificación y Compresión:\*\***

- Asegurar que todos los archivos JS estén minificados (Terser) y servidos con Brotli/Gzip.

---

## ## 3. Optimización del Renderizado Crítico (Impacto: Alto)

**\*\*Problema:\*\*** CSS y scripts bloquean el renderizado inicial.

- **\*\*Acciones Inmediatas:\*\***

### 1. **\*\*CSS Crítico Inline:\*\***

- Extraer el CSS necesario para el above-the-fold e inlinearlo en el ``<head>``. El resto cargar de forma asíncrona:

```
...html
<style>/* CSS crítico */</style>
<link rel="preload" href="styles.css" as="style"
onload="this.onload=null;this.rel='stylesheet'">
...
```

### 2. **\*\*Diferir Scripts No Críticos:\*\***

- Añadir `defer` a scripts propios y `async` a terceros si no dependen del DOM.  
- Mover scripts al final del ``<body>`` cuando sea posible.

### 3. **\*\*Preconnect a Orígenes Externos:\*\***

- Añadir en el ``<head>``:

```
...html
<link rel="preconnect" href="https://www.youtube.com">
<link rel="preconnect" href="https://fonts.gstatic.com">
...
```

---

## ## 4. Mejoras de Caché y Red (Impacto: Medio)

**\*\*Problema:\*\*** Tiempos de caché cortos para recursos de terceros.

- **\*\*Acciones Inmediatas:\*\***

### 1. **\*\*Extender Cache TTL:\*\***

- Configurar `Cache-Control: max-age=31536000, immutable` para recursos estáticos (imágenes, JS, CSS).

### 2. **\*\*Service Worker para Offline:\*\***

- Implementar un service worker con Workbox para cachear activos estáticos y mejorar la carga en visitas recurrentes.

---

### ## 5. Accesibilidad y UX Móvil (Impacto: Medio)

**Problema:** Botones sin nombres accesibles, contraste insuficiente y áreas táctiles pequeñas.

**Acciones Inmediatas:**

1. **Añadir `aria-label` a Botones:**

- Ejemplo: `<button aria-label="Ver proyecto 1">...</button>`.

2. **Mejorar Contraste:**

- Usar una herramienta como WebAIM Contrast Checker para asegurar ratios  $\geq 4.5:1$  en textos normales.

3. **Áreas Táctiles:**

- Asegurar que los botones tengan al menos 48x48 px o espaciado suficiente.

---

### ## 6. Seguridad (Impacto: Bajo)

**Problema:** CSP débil y sin Trusted Types.

**Acciones Recomendadas:**

1. **Refuerzo de CSP:**

- Usar nonces o hashes en lugar de `unsafe-inline`. Ejemplo:

```http

Content-Security-Policy: script-src 'nonce-<RANDOM>' 'strict-dynamic';

```

2. **Trusted Types:**

- Añadir `require-trusted-types-for 'script'` a la CSP.

---

### ## Plan de Implementación

1. **Fase 1 (Crítico, 1-2 días):**

- Redimensionar y comprimir imágenes.

- Implementar carga diferida de YouTube.

- Optimizar CSS crítico y diferir scripts.

2. **Fase 2 (Intermedio, 3-5 días):**

- Configurar caché y service worker.

- Corregir accesibilidad (botones, contraste).

3. **Fase 3 (Avanzado, 1 semana):**

- Reforzar CSP y Trusted Types.

- Monitorear con Lighthouse y WebPageTest.

---

### ## Métricas Esperadas Tras la Optimización

**Móvil:** FCP < 1.5s, LCP < 2.5s, TBT < 200 ms, CLS  $\approx 0$ , Puntuación de Rendimiento >

90.

- \*\*Escritorio:\*\* FCP < 0.8s, LCP < 1.5s, TBT < 100 ms, Puntuación de Rendimiento > 95.

\*\*Nota:\*\* Validar cambios en un entorno de staging antes de producción. Usar herramientas como Lighthouse CI para automatizar pruebas.