



INSTITUTO TECNOLÓGICO DEL PUTUMAYO



IES Vigilada por:
Educación

PROGRAMA INGENIERIA DE SISTEMAS

Página 1

DOCUMENTACION HYPERAGENDADOS

OSCAR DAVID RUIZ SAEZ

KELLY TATIANA ANGULO

WILLIAM DE JESÚS SANTOS CALU

INSTITUTO TECNOLÓGICO DEL PUTUMAYO
MOCOA – PUTUMAYO
2025



PROGRAMA INGENIERIA DE SISTEMAS

Página 2

Tabla de contenido

INTRODUCCION.....	4
Objetivos de HyperAgendados.....	4
Beneficios para la Sociedad	4
¿Cómo Funciona?	5
El Futuro de la Gestión de Salud	5
OBJETIVOS	6
STACK TEGNOGICO	6
Framework principal.....	6
Ventajas en nuestro Stack	6
Scripts	6
Librerías y Dependencias	7
UI y Diseño.....	7
Buenas Prácticas Frontend	8
Accesibilidad Web (WCAG 2.1 Compliance)	10
SEMANTICA HTML.....	11
Estructura Semántica General	11
Jerarquía de Encabezados.....	12
Formularios Semánticos	12
Metadatos y SEO	13
MONITOREO Y ERROR TRACKING.....	15
Lighthouse antes de la instalación de sentry	15
Lighthouse en el login.....	15
Lighthouse en admin.....	15
Lighthouse en empleado.....	16
Lighthouse de usuario	17
Instalación e implementación de sentry	17
Resultados después de implementar sentry	23
Lighthouse en el login.....	23
Lighthouse en admin.....	23



IES Vigilada por:
Educación

INSTITUTO TECNOLÓGICO DEL PUTUMAYO

PROGRAMA INGENIERIA DE SISTEMAS

Página 3

Lighthouse empleado	24
Lighthouse usuario	25
Comparativa de resultados	25

PROGRAMA INGENIERIA DE SISTEMAS

Página 4

INTRODUCCION

HIPERAGENDADOS

¿Qué es HyperAgendados?

HyperAgendados es una innovadora plataforma de gestión de citas médicas diseñada para optimizar la administración de los turnos en centros de salud, hospitales y clínicas. Con un enfoque en la accesibilidad, eficiencia y facilidad de uso, HyperAgendados simplifica la programación y seguimiento de citas tanto para pacientes como para profesionales de la salud.

Objetivos de HyperAgendados

HyperAgendados nace con la misión de transformar la manera en que se gestionan las citas médicas, resolviendo problemas comunes como la desorganización, largas esperas y la falta de transparencia en la asignación de turnos. Sus principales objetivos son:

- **Reducir tiempos de espera:** Los pacientes pueden acceder fácilmente a turnos disponibles sin necesidad de acudir físicamente a un centro de salud.
- **Optimizar el uso del tiempo de los profesionales de la salud:** Los médicos y empleados pueden gestionar su agenda de manera más eficiente, reduciendo cancelaciones y espacios perdidos.
- **Mejorar la accesibilidad y la inclusión:** Una interfaz accesible y optimizada permite a cualquier usuario, independientemente de su edad o nivel de conocimiento tecnológico, agendar citas de manera sencilla.
- **Brindar una gestión segura y transparente:** La plataforma permite a los administradores tener un control completo sobre las citas, reasignaciones y usuarios registrados.
- **Facilitar la atención médica:** Con una organización estructurada de los turnos y especialidades, se agiliza el acceso a la atención médica adecuada.

Beneficios para la Sociedad

HyperAgendados no es solo una herramienta para agendar citas; es una solución integral que mejora significativamente el acceso a la salud pública y privada. Sus beneficios clave incluyen:

- **Atención médica equitativa y eficiente:** Todos los pacientes tienen la misma oportunidad de acceder a un turno sin importar su ubicación o condición económica.
- **Eliminación de barreras geográficas:** Pacientes pueden consultar disponibilidad de citas desde cualquier lugar y seleccionar la opción que mejor les convenga.
- **Reducción de costos administrativos:** Disminuye la carga de trabajo manual en hospitales y clínicas, permitiendo un mejor uso de los recursos.

PROGRAMA INGENIERIA DE SISTEMAS

Página 5

- **Mayor control y prevención en salud:** Con un sistema organizado y accesible, se reducen las tasas de inasistencia y se fomenta una atención preventiva más efectiva.
- **Eficiencia en la asignación de recursos médicos:** La plataforma permite gestionar y reasignar citas con rapidez, evitando saturaciones en consultorios.

¿Cómo Funciona?

HyperAgendados está diseñado con una estructura intuitiva y fácil de usar para tres tipos de usuarios: **administradores, empleados y pacientes.**

1. Pacientes:

- Ingresan a la plataforma y seleccionan su especialidad de interés.
- Ven un listado de citas disponibles y pueden agendarse con un solo clic.
- Reciben confirmaciones y recordatorios automáticos.

2. Empleados de Salud:

- Visualizan y gestionan su agenda de citas.
- Pueden registrar nuevas citas disponibles para los pacientes.
- Cuentan con un sistema seguro que les permite mantener un orden estricto en su trabajo.

3. Administradores:

- Tienen control total sobre la plataforma.
- Pueden gestionar citas, asignarlas, reasignarlas y cancelarlas según disponibilidad y necesidad.
- Administran a los usuarios, gestionan roles y aseguran un funcionamiento eficiente del sistema.

El Futuro de la Gestión de Salud

HyperAgendados no es solo una plataforma de citas, es la evolución hacia un sistema de salud más organizado, eficiente y accesible para todos. Con su implementación, hospitales y clínicas pueden mejorar drásticamente su capacidad de atención, reduciendo tiempos de espera y optimizando el trabajo de los profesionales de la salud.

Esta solución representa un gran paso hacia la digitalización médica, garantizando que ningún paciente se quede sin atención y que cada cita se aproveche al máximo. HyperAgendados es la clave para una gestión de salud moderna, funcional y orientada al bienestar de la sociedad.

GitHub: <https://github.com/Rudzy302/HYPERAGENDADOS>

Despliegue: <https://hyperagendados.vercel.app>

PROGRAMA INGENIERIA DE SISTEMAS

Página 6

OBJETIVOS

Los objetivos a cumplir en hyperagendados son:

- Implementar las buenas prácticas para el desarrollo web frontend visto en la materia
- Garantizar la accesibilidad para los usuarios con discapacidades físicas
- Optimizar el rendimiento y las métricas web esenciales para el desarrollo
- Monitoreo y tracking de errores en producción
- Cumplir con los estándares de seguridad web

STACK TECNOLÓGICO

Framework principal

- **Next.js V 15.3.3** renderizado híbridamente con react
- **React V 19.0.0** biblioteca para las interfaces de usuarios
- **TypeScript 5** Tipación de carácter estático para Javascript

Ventajas en nuestro Stack

- **Server-Side Rendering(SSR)** para SEO
- **Static Site Generation (SSG)** para las páginas estáticas
- **Hot Module Replacement** para el desarrollo eficiente
- **Optimización automática** de imágenes y bundles
- **Tipado estático** para mayor robusticidad en nuestro código
- **Tailwind** para los estilos dentro de las páginas
- **.ENV** para el manejo de rutas y conexión directa con su backend con su base de datos

Scripts

- **"dev": "next dev --turbo"** → Inicia Next.js en modo desarrollo utilizando Turbopack, un motor que mejora la velocidad de construcción y la recarga en caliente, permitiendo cambios rápidos en el código sin perder rendimiento.
- **"build": "next build"** → Compila y optimiza la aplicación para producción. Este proceso genera un conjunto de archivos listos para ser desplegados en un servidor, asegurando un rendimiento óptimo.
- **"start": "next start"** → Ejecuta la aplicación en modo producción después de haber realizado la compilación con `next build`. Es el comando que se usa para poner el proyecto en funcionamiento real.

PROGRAMA INGENIERIA DE SISTEMAS

Página 7

- **"lint": "next lint"** → Analiza el código con ESLint para detectar errores y mejorar la calidad del desarrollo. Ayuda a mantener un código limpio, organizado y libre de problemas comunes.

Librerías y Dependencias

UI y Diseño

- **Next.js (v15.3.3)**: Framework de React para la construcción de aplicaciones web de alto rendimiento, con renderizado híbrido (SSR, SSG, CSR) y routing basado en archivos. Es la base de la aplicación.
- **Tailwind CSS (v4)**: Framework CSS utility-first para un desarrollo rápido y un diseño altamente personalizable y responsivo.
- **React Icons (v5.5.0)**: Librería de iconos que ofrece una amplia gama de sets de iconos populares para una iconografía consistente.
- **Framer Motion (v12.6.3)**: Librería para animaciones fluidas y declarativas en React, utilizada para mejorar la experiencia de usuario con transiciones y microinteracciones.
- **Otros:**
 - @tailwindcss/postcss: Integración de Tailwind con PostCSS.
 - clsx: Utilidad para construir cadenas de nombres de clase condicionalmente.

Autenticación y Seguridad

- **Next-Auth (v4.24.11)**: Solución completa para la autenticación en aplicaciones Next.js, compatible con OAuth, JWT y gestión de sesiones segura.
- **Zod (v3.23.8)**: Librería para la validación de esquemas (schemas), lo que permite definir y validar estructuras de datos de forma type-safe. Principalmente útil en formularios y validaciones de entrada.

Gestión de Estado y Formularios

- **React Hook Form (v7.54.2)**: Librería optimizada para la gestión de formularios en React, ofreciendo rendimiento y un manejo sencillo de la validación.

Monitoreo y Analytics

- **Sentry (con @sentry/nextjs v9.27.0, @sentry/react v9.27.0, @sentry/cli v2.46.0)**: Plataforma de monitoreo de errores y rendimiento. Integrada para capturar errores tanto del lado del cliente como del servidor/Edge, y para subir sourcemaps, facilitando la depuración en producción.

Herramientas de Desarrollo

- **TypeScript (v5)**: Lenguaje de programación superset de JavaScript que añade tipado estático, mejorando la robustez y mantenibilidad del código.

PROGRAMA INGENIERIA DE SISTEMAS

Página 8

- **ESLint (v9):** Herramienta de linting para identificar y reportar patrones problemáticos en el código JavaScript/TypeScript, asegurando la consistencia y calidad del código.
- **@types/node, @types/react, @types/react-dom:** Definiciones de tipos para Node.js, React y React DOM, esenciales para el desarrollo con TypeScript.
- **eslint-config-next:** Configuración de ESLint específica para Next.js.

Buenas Prácticas Frontend

El desarrollo del frontend de HYPERAGENDADOS sigue las siguientes buenas prácticas para asegurar un código limpio, mantenible y escalable:

Arquitectura de Componentes

La estructura de directorios (src/components/) organiza los componentes de manera lógica:

- **components/ui/:** Contiene componentes base reutilizables (ej. Input, Button) que no están directamente relacionados con la lógica de negocio y pueden ser compartidos a través de la aplicación.
- **components/Shared/:** Aloja componentes que encapsulan lógica de negocio específica y son reutilizados en diferentes secciones de la aplicación.
- **components/auth/dashboard/:** Contiene los componentes específicos de las vistas del dashboard para cada rol (admin, empleado, usuario).

Principios SOLID Aplicados

Se busca aplicar los principios SOLID en la medida de lo posible:

- **Single Responsibility Principle (SRP):** Cada componente y módulo se diseña con una única responsabilidad clara, mejorando la legibilidad y facilitando el mantenimiento.
- **Open/Closed Principle (OCP):** Los componentes están diseñados para ser extensibles mediante props, permitiendo añadir nueva funcionalidad sin modificar su código fuente principal.
- **Dependency Inversion Principle (DIP):** Se fomenta el uso de hooks personalizados para encapsular la lógica de negocio, desacoplando los componentes de su implementación.

Gestión de Estado Inmutable

- El estado de los componentes se gestiona utilizando useState y useReducer de React.
- Las actualizaciones de estado se realizan de forma inmutable para evitar efectos secundarios y facilitar la depuración.
- La validación de datos con Zod contribuye a la seguridad de tipos y la consistencia del estado.

PROGRAMA INGENIERIA DE SISTEMAS

Página 9

Code Splitting y Lazy Loading

- Next.js implementa automáticamente el code splitting.
- Se utiliza dynamic (de next/dynamic) para la carga perezosa (lazy loading) de componentes pesados o que no son críticos para la carga inicial de la página, mejorando el tiempo de carga de la aplicación.

TypeScript

// Ejemplo de implementación de lazy loading

import dynamic from 'next/dynamic';

```
const DynamicComponent = dynamic(() => import('./HeavyComponent'), {  
  loading: () => <p>Cargando...</p>,  
  ssr: false // No renderizar en el servidor si no es necesario  
});
```

Error Boundaries

- Se utiliza un error.tsx global (o global-error.tsx) para capturar errores en cascada en los componentes de React, previniendo que toda la aplicación se caiga y ofreciendo una interfaz de usuario de recuperación.

TypeScript

// src/app/global-error.tsx (Ejemplo de Error Boundary global)

'use client';

import React from 'react'; // Asegúrate de importar React

```
export default function GlobalError({ error, reset }: {  
  error: Error & { digest?: string };  
  reset: () => void;  
}) {  
  return (  
    <html>  
      <body>  
        <h2>¡Algo salió mal!</h2>  
        <button onClick={() => reset()}>Intentar de nuevo</button>  
      </body>  
    </html>  
  );  
}
```

PROGRAMA INGENIERIA DE SISTEMAS

Página 10

Accesibilidad Web (WCAG 2.1 Compliance)

El frontend está diseñado para cumplir con las pautas de accesibilidad WCAG 2.1 nivel AA, asegurando que la aplicación sea usable por personas con diversas discapacidades.

Principios WCAG

- **Perceivable (Perceptible):**
 - **Contraste de color:** Ratio mínimo de 4.5:1 para texto normal, garantizando la legibilidad.
 - **Texto alternativo:** Todas las imágenes y elementos no textuales interactivos tendrán atributos alt descriptivos.
 - **Contenido multimedia:** (Aplicaría si hubiera videos o audios, con subtítulos y transcripciones).
- **Operable (Operable):**
 - **Navegación por teclado:** Todos los elementos interactivos son accesibles y utilizables mediante el teclado (tabulación, enter, espacio).
 - **Focus management:** Indicadores de foco visibles y lógicos para orientar a los usuarios de teclado.
 - **Tiempo suficiente:** No se imponen límites de tiempo automáticos críticos que impidan la interacción del usuario.
- **Understandable (Comprensible):**
 - **Idioma de página:** El atributo lang se declara correctamente en el HTML.
 - **Labels descriptivos:** Los formularios utilizan etiquetas claras y asociadas correctamente (<label htmlFor="id">).
 - **Mensajes de error:** Se proporciona feedback claro y específico para los errores de validación o del sistema.
- **Robust (Robusto):**
 - **HTML válido:** El código HTML generado es semánticamente correcto y sigue los estándares.
 - **Compatibilidad:** La aplicación funciona con tecnologías asistivas (lectores de pantalla, lupas, etc.).

Implementación de Accesibilidad

- **Uso de aria-* attributes:** Se utilizan atributos WAI-ARIA donde es necesario para mejorar la semántica y la información para tecnologías asistivas.
- **Landmarks semánticos:** Se emplean elementos HTML semánticos (<main>, <nav>, <aside>, <footer>, <header>) para estructurar el contenido y facilitar la navegación por parte de los lectores de pantalla.
- **Live regions:** Se consideran regiones en vivo para proporcionar feedback dinámico a los usuarios de lectores de pantalla (ej. mensajes de éxito/error después de una acción).
- **Keyboard Navigation:**
 - Se asegura un orden de tabulación lógico.

PROGRAMA INGENIERIA DE SISTEMAS

Página 11

- Se maneja la tecla Escape para cerrar modales o menús.
- Se permite interactuar con elementos clickeables usando Enter o Space.
- **ToolTip:** Se implementa para dar al usuario una descripción de que hace cada botón

SEMANTICA HTML

Estructura Semántica General

Los componentes de la aplicación se construyen utilizando elementos HTML que transmiten el propósito y la función de cada sección del contenido.

- **Ejemplos clave vistos en HYPERAGENDADOS:**
 - Uso de <header> y <footer> para las secciones de cabecera y pie de página globales o de módulos específicos, proporcionando contexto.
 - La navegación principal del sistema de roles (Paciente, Empleado, Administrador) se encapsula en un elemento <nav>.
 - El contenido principal y único de cada vista (ej., el dashboard del administrador, el formulario de agendamiento) se define dentro de un elemento <main>.
 - Secciones temáticas dentro de una página (ej., "Próximas Citas", "Historial de Pacientes", "Formulario de Búsqueda") se agrupan con el elemento <section>, a menudo complementado con un aria-labelledby para asociarlo a un encabezado visible.

HTML

```
<header>
<h1>HYPERAGENDADOS - Panel de Empleado</h1>
<nav aria-label="Navegación del panel de empleado">
  <ul>
    <li><a href="/empleado/agenda">Mi Agenda</a></li>
    <li><a href="/empleado/citas-disponibles">Citas Disponibles</a></li>
    <li><a href="/empleado/pacientes">Gestión de Pacientes</a></li>
  </ul>
</nav>
</header>
<main>
<section aria-labelledby="citas-pendientes-heading">
  <h2 id="citas-pendientes-heading">Citas Pendientes para Hoy</h2>
  <table>
    <thead>
      <tr>
        <th>Hora</th>
        <th>Paciente</th>
        <th>Servicio</th>
      </tr>
    </thead>
```

PROGRAMA INGENIERIA DE SISTEMAS

Página 12

```
<tbody>
<tr>
<td>09:00 AM</td>
<td>Juan Pérez</td>
<td>Consulta General</td>
</tr>
</tbody>
</table>
</section>
<section aria-labelledby="acciones-rapidas-heading">
<h2 id="acciones-rapidas-heading">Acciones Rápidas</h2>
<button type="button">Registrar Nueva Cita</button>
<button type="button">Ver Mi Calendario Completo</button>
</section>
</main>
<footer>
<p>&copy; 2025 HYPERAGENDADOS. Todos los derechos reservados.</p>
</footer>
```

Jerarquía de Encabezados

Se aplica una jerarquía estricta y lógica de encabezados (<h1> a <h6>) para organizar la información en cada página, mejorando la legibilidad y la navegabilidad.

- **Un único <h1> por página:** Cada vista principal del frontend de HYPERAGENDADOS posee un solo <h1> que actúa como el título principal de la página, indicando claramente su propósito.
 - En /dashboard/admin: <h1>Panel de Administración de HYPERAGENDADOS</h1>
 - En /agendar-cita: <h1>Agendar Nueva Cita Médica</h1>
 - En /perfil: <h1>Mi Perfil de Usuario</h1>
- **Jerarquía lógica y secuencial:** Los encabezados subsiguientes (<h2>, <h3>, etc.) se utilizan de forma anidada y lógica para estructurar el contenido en secciones y subsecciones, siguiendo un orden de importancia.
 - Si <h1> es "Agendar Nueva Cita Médica", entonces <h2> podría ser "Selecciona tu Especialidad", <h3> "Elige un Profesional Disponible", y <h4> "Confirma tu Horario".
 - En el "Panel de Administración", <h2> podría ser "Gestión de Empleados", con <h3> "Listado de Empleados" y otro <h3> "Añadir Nuevo Empleado".

Formularios Semánticos

Los formularios en HYPERAGENDADOS se construyen con elementos HTML semánticos y asociaciones correctas para garantizar la accesibilidad, una entrada de datos clara y una experiencia de usuario intuitiva.

PROGRAMA INGENIERIA DE SISTEMAS

Página 13

- **Elementos <label>:** Cada campo de entrada (<input>, <select>, <textarea>) está asociado explícitamente a un elemento <label> mediante los atributos htmlFor e id. Esto permite que los lectores de pantalla anuncien la etiqueta correcta para cada campo.
- **Controles de formulario apropiados:** Se utilizan los tipos de <input> más adecuados (type="email", type="date", type="tel", type="password") para aprovechar la validación del navegador y el teclado virtual en dispositivos móviles.
- **Mensajes de error accesibles:** Los mensajes de error de validación son visibles y se asocian con los campos de formulario relevantes utilizando aria-describedby y role="alert", para que las tecnologías asistivas puedan anunciar estos errores al usuario.

HTML

```
<form aria-label="Buscador de citas médicas">
  <fieldset>
    <legend>Buscar Citas Disponibles</legend>

    <div>
      <label htmlFor="especialidad">Especialidad:</label>
      <select id="especialidad" name="especialidad" required aria-describedby="especialidad-
error">
        <option value="">-- Seleccione una especialidad --</option>
        <option value="cardiologia">Cardiología</option>
        <option value="dermatologia">Dermatología</option>
        <option value="pediatria">Pediatria</option>
      </select>
      <p id="especialidad-error" role="alert" style="color: red;">Debe seleccionar una
especialidad.</p>
    </div>

    <div>
      <label htmlFor="fechaCita">Fecha Deseada:</label>
      <input type="date" id="fechaCita" name="fechaCita" required min={ new
Date().toISOString().split("T")[0]} />
    </div>

    <button type="submit">Buscar Citas</button>
  </fieldset>
</form>
```

Metadatos y SEO

La gestión de metadatos en HYPERAGENDADOS se implementa para optimizar la visibilidad en motores de búsqueda (SEO) y controlar la presentación de la información al ser compartida en redes sociales. Estos metadatos se materializan como etiquetas HTML dentro de la sección <head> de cada página generada.

PROGRAMA INGENIERIA DE SISTEMAS

Página 14

Metadatos en el HTML de la página principal de registro/landing de HYPERAGENDADOS:**HTML**

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>HYPERAGENDADOS - Tu Plataforma de Gestión de Citas Médicas</title>

  <meta name="description" content="Optimiza la gestión de tus citas médicas y de salud.
  HYPERAGENDADOS es la plataforma eficiente para pacientes, empleados y administradores
  de centros de salud." />

  <meta name="keywords" content="gestión de citas, citas médicas, salud digital, agendamiento
  online, clínicas, hospitales, pacientes, profesionales de la salud, HYPERAGENDADOS" />

  <meta property="og:title" content="HYPERAGENDADOS: Gestión de Citas Médicas
  Simplificada" />
  <meta property="og:description" content="Descubre la forma más eficiente de agendar y
  administrar citas médicas para centros de salud." />
  <meta property="og:url" content="https://www.hyperagendados.com/" />
  <meta property="og:site_name" content="HYPERAGENDADOS" />
  <meta property="og:image" content="https://www.hyperagendados.com/imagenes/og-inicio-
  hyperagendados.png" />
  <meta property="og:image:width" content="1200" />
  <meta property="og:image:height" content="630" />
  <meta property="og:type" content="website" />

  <meta name="twitter:card" content="summary_large_image" />
  <meta name="twitter:site" content="@HyperAgendadosOficial" /> <meta
  name="twitter:title" content="Revoluciona tus Citas con HYPERAGENDADOS" />
  <meta name="twitter:description" content="Software de gestión de citas médicas para una
  salud más organizada y accesible." />
  <meta name="twitter:image" content="https://www.hyperagendados.com/imagenes/twitter-
  card-inicio.png" />

  <link rel="canonical" href="https://www.hyperagendados.com/" />

  <meta name="robots" content="index, follow" />

  <link rel="icon" href="/favicon.ico" />
```


PROGRAMA INGENIERIA DE SISTEMAS

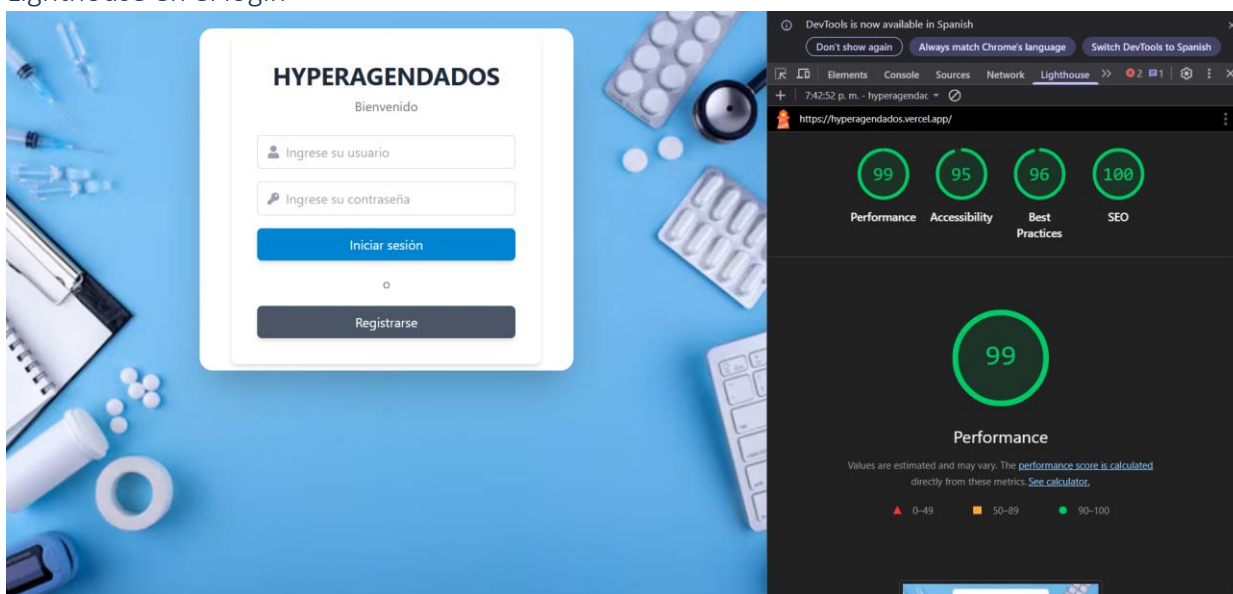
Página 15

```
</head>
<body>
  <div id="__next">
    </div>
  </body>
</html>
```

MONITOREO Y ERROR TRACKING

Lighthouse antes de la instalación de sentry

Lighthouse en el login



- **First Contentful Paint** “primera Pintura con Contenido”: 0.3 segundos
- **Largest Contentful Paint** “pintura de contenido más grande”: 0.5 segundos
- **Total Blocking Time** “tiempo total de bloqueo”: 300 milisegundos
- **Cumulative Layout Shift** “cambio acumulativo de diseño” 0 segundos
- **Speed Index** “índice de la velocidad: 0.8 segundos

Lighthouse en admin

PROGRAMA INGENIERIA DE SISTEMAS

Página 16

HYPERAGENDADOS

Panel de Gestión para Administradores

Panel de Administrador

Cerrar Sesión

Crear Nuevo Usuario

Nombre Completo

Correo Electrónico

Contraseña

Seleccionar Rol

Crear Usuario

Gestión de Usuarios y Citas

NOMBRE	ROL	CITAS ASIGNADAS	ACCIONES
Juan Pérez (Admin)	Administrador	5 citas	Editar Gestionar Eliminar
María López (Emp)	Empleado	12 citas	Editar Gestionar Eliminar

DevTools is now available in Spanish

Don't show again Always match Chrome's language Switch DevTools to Spanish

Elements Console Sources Lighthouse

11:43:53 - hyperagendados.v

https://hyperagendados.vercel.app/dashboard

Performance Accessibility Best Practices SEO

91

What's new in DevTools 137

See all new features

See the highlights from Chrome 137

- **First Contentful Paint** “primera Pintura con Contenido”: 1 segundos
- **Largest Contentful Paint** “pintura de contenido más grande”: 02.1 segundos
- **Total Blocking Time** “tiempo total de bloqueo”: 350 milisegundos
- **Cumulative Layout Shift** “cambio acumulativo de diseño” 0 segundos
- **Speed Index** “índice de la velocidad: 1.9 segundos

Lighthouse en empleado

HYPERAGENDADOS

Panel de Gestión para Empleados

Panel de Empleado

Cerrar Sesión

Crear Nueva Cita

dd/mm/aaaa

Profesión: [Profesión del Empleado Automática]

Crear Cita

La profesión del empleado que crea la cita se asignará automáticamente.

Mis Citas (Disponibles y Agendadas)

Cita Disponible

Profesión: Pediatría

Fecha: 25/07/2025

Hora: 11:00 AM

Estado: Disponible

Cita Agendada

Profesión: Pediatría

Fecha: 20/06/2025

Hora: 09:30 AM

Estado: Agendada

DevTools is now available in Spanish

Don't show again Always match Chrome's language Switch DevTools to Spanish

Elements Console Sources Lighthouse

11:47:27 - hyperagendados.v

https://hyperagendados.vercel.app/dashboard

Performance Accessibility Best Practices SEO

95

What's new in DevTools 137

See all new features

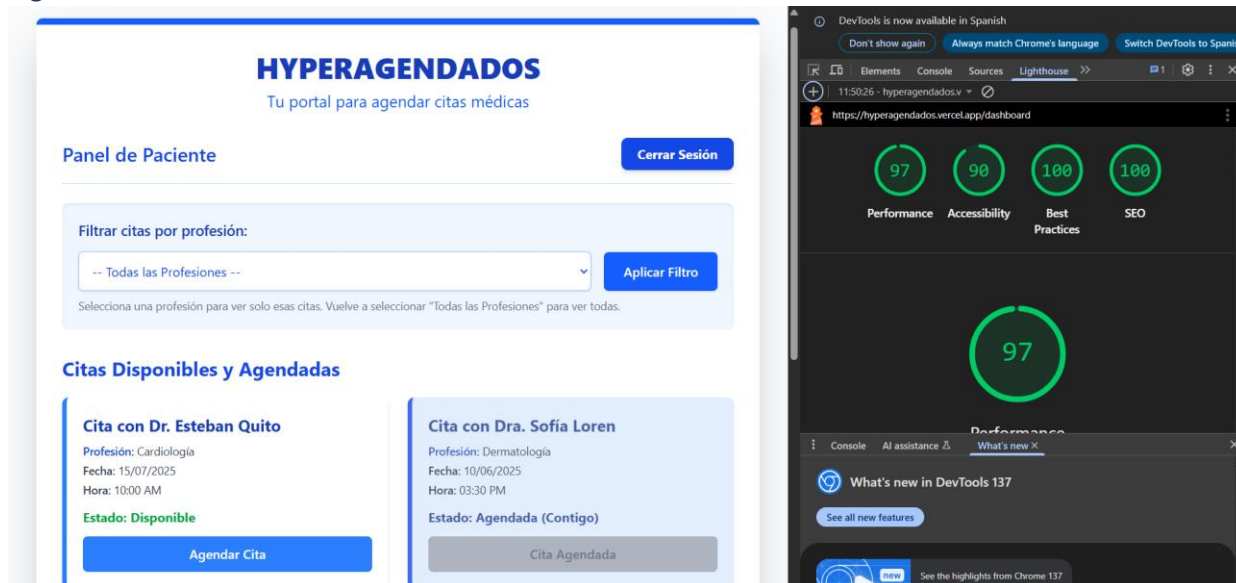
See the highlights from Chrome 137

PROGRAMA INGENIERIA DE SISTEMAS

Página 17

- **First Contentful Paint** “primera Pintura con Contenido”: 0.1 segundos
- **Largest Contentful Paint** “pintura de contenido más grande”: 2 segundos
- **Total Blocking Time** “tiempo total de bloqueo”: 230 milisegundos
- **Cumulative Layout Shift** “cambio acumulativo de diseño” 0 segundos
- **Speed Index** “índice de la velocidad: 1.5 segundos

Lighthouse de usuario



- **First Contentful Paint** “primera Pintura con Contenido”: 0.9 segundos
- **Largest Contentful Paint** “pintura de contenido más grande”: 1.9 segundos
- **Total Blocking Time** “tiempo total de bloqueo”: 180 milisegundos
- **Cumulative Layout Shift** “cambio acumulativo de diseño” 0 segundos
- **Speed Index** “índice de la velocidad: 1.7 segundos

Instalación e implementación de sentry

El primer paso es instalar los paquetes de Sentry, en este caso el comando usado será “*npm install @sentry/nextjs @sentry/react @sentry/node*”.

PROGRAMA INGENIERIA DE SISTEMAS

Página 18

```
$ npm install @sentry/nextjs @sentry/react @sentry/node  
  
added 618 packages, and audited 619 packages in 1m  
  
152 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities
```

Una vez instalados, configuraremos Sentry para el lado del cliente (navegador) y para el lado del servidor (Node.js) en tu aplicación Next.js, esto implica crear dos archivos de configuración: `sentry.client.config.ts` y `sentry.server.config.ts` en la raíz del directorio frontend.

`sentry.client.config.ts`

```
1  import * as Sentry from "@sentry/nextjs";  
2  
3  Sentry.init({  
4    dsn: process.env.NEXT_PUBLIC_SENTRY_DSN,  
5    integrations: [  
6      Sentry.browserTracingIntegration(),  
7      Sentry.replayIntegration({  
8        maskAllInputs: true,  
9        blockAllMedia: true,  
10     })),  
11   ],  
12   tracesSampleRate: 1.0,  
13   replaysSessionSampleRate: 0.1,  
14   replaysOnErrorSampleRate: 1.0,  
15 });  
16
```

`sentry.server.config.ts`

PROGRAMA INGENIERIA DE SISTEMAS

Página 19

```
1 import * as Sentry from "@sentry/nextjs";
2
3 Sentry.init({
4   dsn: process.env.SENTRY_DSN,
5   tracesSampleRate: 1.0,
6 });
7
```

Se modificó el archivo next.config.ts del frontend para envolver la configuración de Next.js con withSentryConfig.

```
1 import type { NextConfig } from "next";
2 import { withSentryConfig } from "@sentry/nextjs";
3
4 const nextConfig: NextConfig = {
5   /* config options here */
6 };
7
8 const sentryWebpackPluginOptions = {
9   silent: true,
10  org: "propia", // organización de Sentry
11  project: "javascript-nextjs", // proyecto de Sentry
12 };
13
14 export default withSentryConfig(nextConfig, sentryWebpackPluginOptions);
15
```

Posteriormente, modifiqué el archivo “.env” del frontend para almacenar las variables de entorno sensibles (los DSNs de Sentry), “SENTRY_DSN” & “NEXT_PUBLIC_SENTRY_DSN”

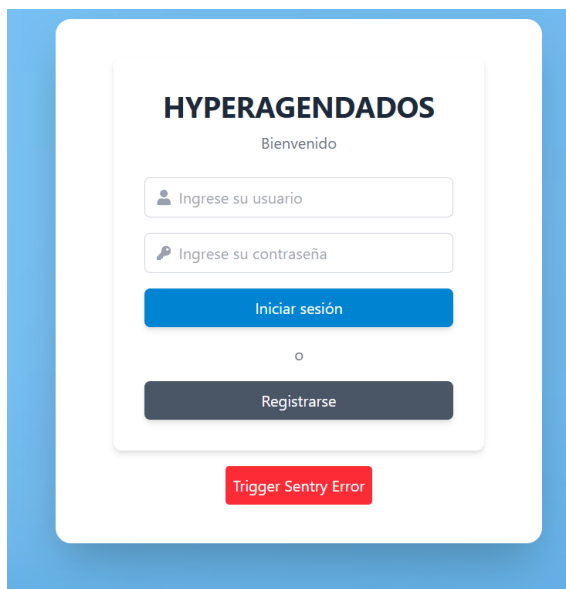
Y finalmente Se modificó el archivo layout.tsx para envolver el contenido principal de la aplicación con un Sentry.ErrorBoundary.


```
1 import SessionAuthProvider from "@context/SessionAuthProvider";
2 import "../globals.css";
3 import type { Metadata } from "next";
4 import * as Sentry from "@sentry/nextjs";
5
6 export const metadata: Metadata = {
7   title: "HyperAgendados",
8   description: "Agendamiento de citas",
9 };
10
11 export default function RootLayout({
12   children,
13 }: {
14   children: React.ReactNode;
15 }) {
16   return (
17     <html lang="es">
18       <body>
19         <Sentry.ErrorBoundary
20           fallback={({ error, resetError }) => (
21             <div>
22               <h2>Ocurrió un error!</h2>
23               <p>{(error as Error).message}</p>
24               <button onClick={() => resetError()}>Intentar de nuevo</button>
25             </div>
26           )}
27         >
28           <SessionAuthProvider>{children}</SessionAuthProvider>
29         </Sentry.ErrorBoundary>
30       </body>
31     </html>
32   );
33 }
34
```

Se configura un botón para provocar un error y que sentry lo detecte, ya que se pide para verificar que sentry este correctamente funcionando.

PROGRAMA INGENIERIA DE SISTEMAS

Página 21



Get Started with Sentry Issues
Your code sleuth eagerly awaits its first mission.

Set up the Sentry SDK

- 1 Install Sentry ✓
- 2 Configure Sentry ✓
- 3 Verify

Start your development server and visit `/sentry-example-page` if you have set it up. Click the button to trigger a test error.

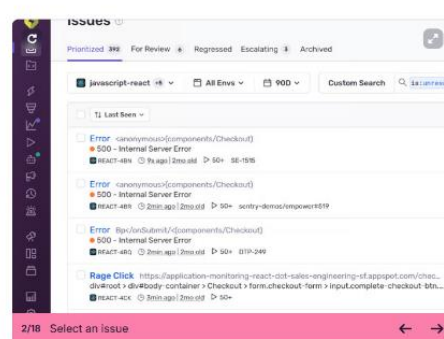
Or, trigger a sample error by calling a function that does not exist somewhere in your application.

```
JavaScript  
myUndefinedFunction();
```

Event was received!

[Back](#) [Take me to my error](#)

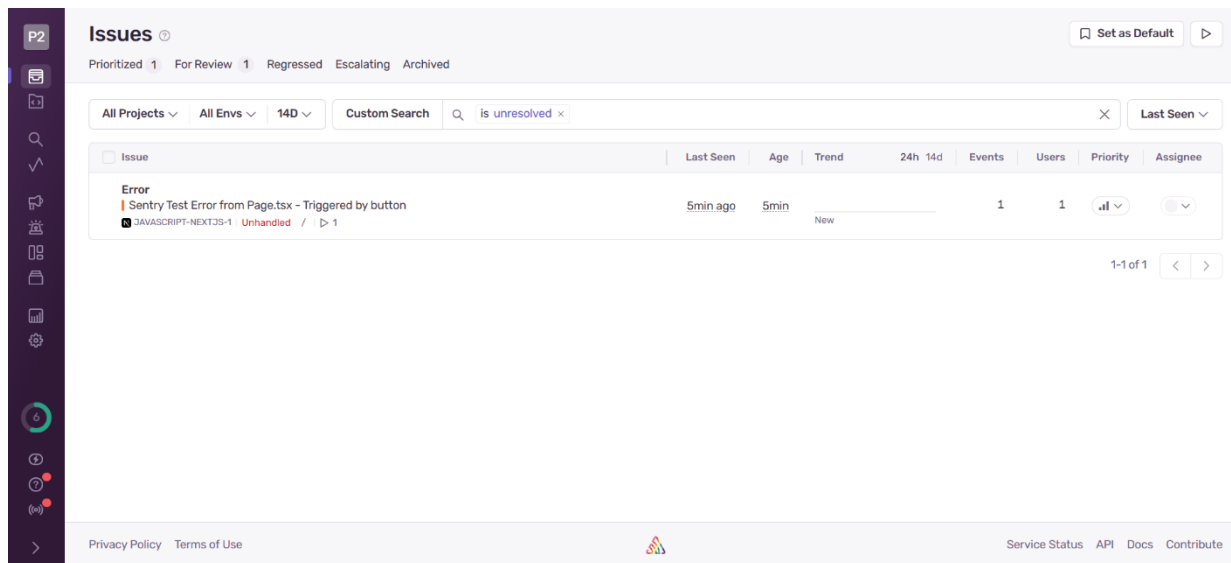
Preview a Sentry Issue



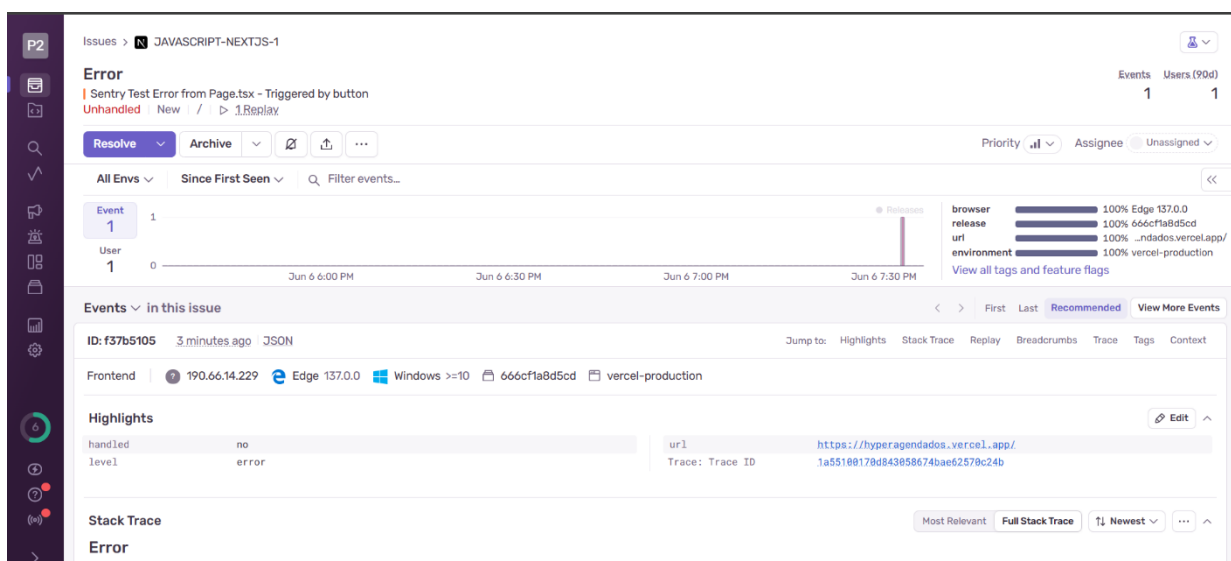
Y ya quedaría Sentry completamente funcional en el frontend.

PROGRAMA INGENIERIA DE SISTEMAS

Página 22



The screenshot shows the Sentry 'Issues' dashboard. At the top, there are filters for 'All Projects', 'All Envs', and '14D'. A search bar contains 'is unresolved'. Below the filters, a table lists issues. The first issue is an 'Error' titled 'Sentry Test Error from Page.tsx - Triggered by button', which is 'Unhandled' and occurred '5min ago'. The table has columns for 'Issue', 'Last Seen', 'Age', 'Trend', '24h 14d', 'Events', 'Users', 'Priority', and 'Assignee'. At the bottom, there are links for 'Privacy Policy', 'Terms of Use', 'Service Status', 'API', 'Docs', and 'Contribute'.



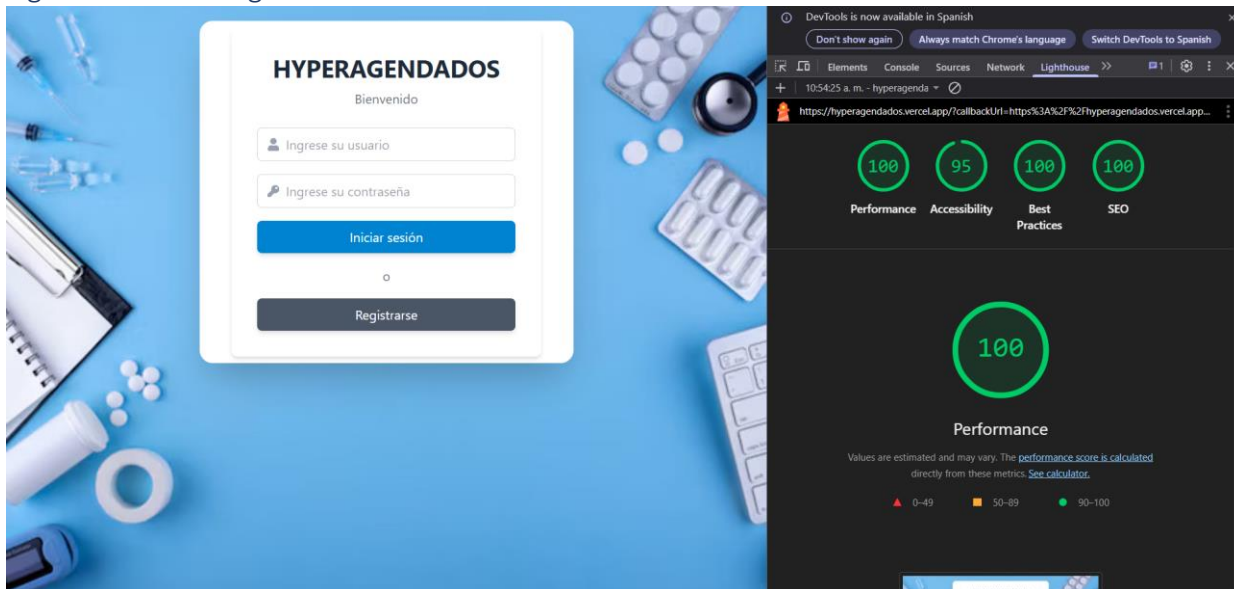
The screenshot shows the details of a specific issue in Sentry. The issue is titled 'Error' and is 'Unhandled'. It shows a timeline of events with a graph. Below the graph, there are tabs for 'Events', 'Releases', 'Browser', 'Environment', and 'Tags'. The 'Events' tab is active, showing a list of events. The first event is 'ID: f37b5105' with a timestamp of '3 minutes ago'. It includes details about the 'Frontend' environment, the 'User' (190.66.14.229), the 'Browser' (Edge 137.0.0), the 'OS' (Windows), the 'Release' (666cf1a8d5cd), and the 'Environment' (vercel-production). The 'Highlights' section shows that the error was 'handled' and the 'level' was 'error'. The 'Stack Trace' section shows the error message: 'Sentry Test Error from Page.tsx - Triggered by button'.

PROGRAMA INGENIERIA DE SISTEMAS

Página 23

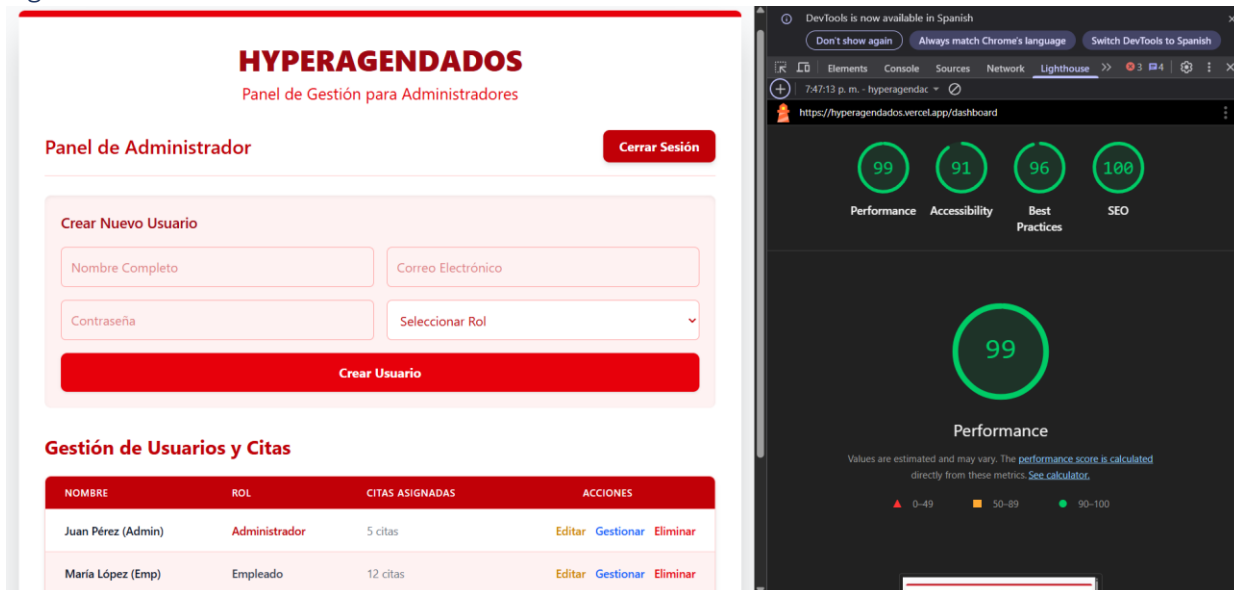
Resultados después de implementar sentry

Lighthouse en el login



- **First Contentful Paint** “primera Pintura con Contenido”: 0.3 segundos
- **Largest Contentful Paint** “pintura de contenido más grande”: 0.5 segundos
- **Total Blocking Time** “tiempo total de bloqueo”: 80 milisegundos
- **Cumulative Layout Shift** “cambio acumulativo de diseño” 0 segundos
- **Speed Index** “índice de la velocidad: 0.8 segundos

Lighthouse en admin



PROGRAMA INGENIERIA DE SISTEMAS

Página 24

- **First Contentful Paint** “primera Pintura con Contenido”: 0.3 segundos
- **Largest Contentful Paint** “pintura de contenido más grande”: 0.9 segundos
- **Total Blocking Time** “tiempo total de bloqueo”: 70 milisegundos
- **Cumulative Layout Shift** “cambio acumulativo de diseño” 0 segundos
- **Speed Index** “índice de la velocidad: 0.9 segundos

Lighthouse empleado



HYPERAGENDADOS
Panel de Gestión para Empleados

Panel de Empleado Cerrar Sesión

Crear Nueva Cita

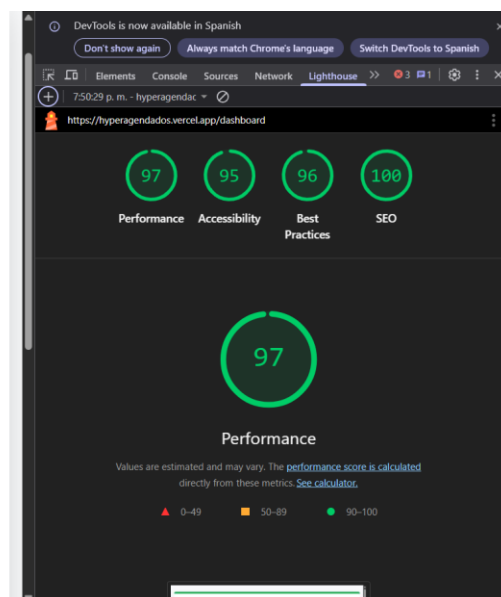
dd/mm/aaaa Profesión: [Profesión del Empleado Automática]

Crear Cita

La profesión del empleado que crea la cita se asignará automáticamente.

Mis Citas (Disponibles y Agendadas)

Cita Disponible	Cita Agendada
Profesión: Pediatría Fecha: 25/07/2025 Hora: 11:00 AM Estado: Disponible	Profesión: Pediatría Fecha: 20/06/2025 Hora: 09:30 AM Estado: Agendada

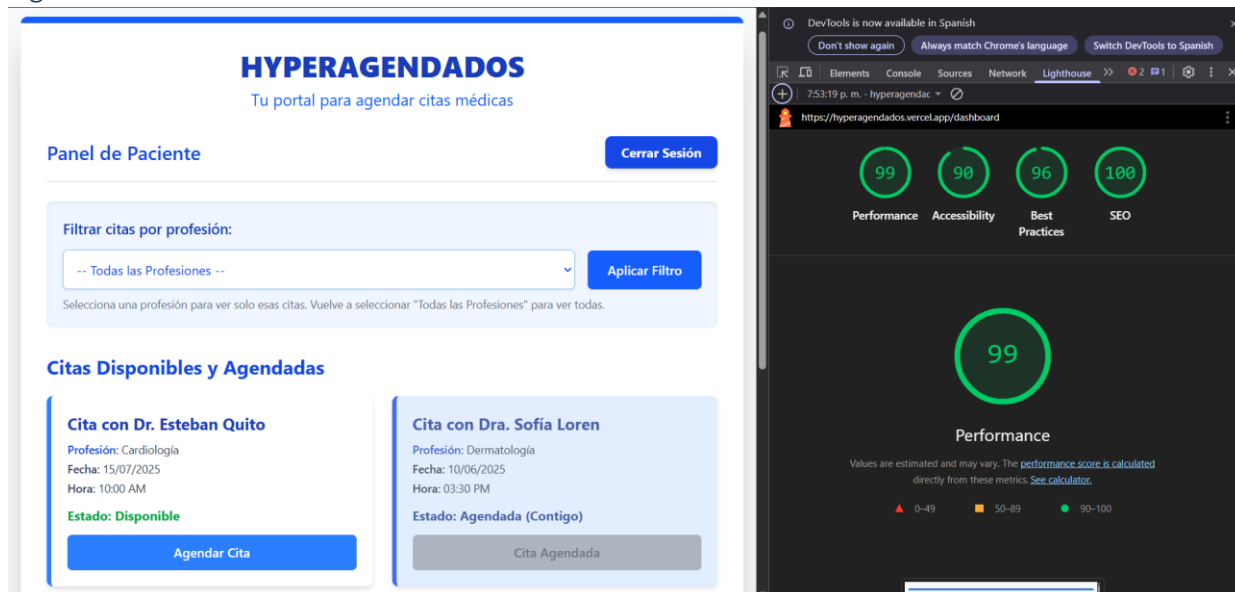


- **First Contentful Paint** “primera Pintura con Contenido”: 0.1 segundos
- **Largest Contentful Paint** “pintura de contenido más grande”: 1 segundos
- **Total Blocking Time** “tiempo total de bloqueo”: 60 milisegundos
- **Cumulative Layout Shift** “cambio acumulativo de diseño” 0 segundos
- **Speed Index** “índice de la velocidad: 0.5 segundos

PROGRAMA INGENIERIA DE SISTEMAS

Página 25

Lighthouse usuario



- **First Contentful Paint** “primera Pintura con Contenido”: 0.3 segundos
- **Largest Contentful Paint** “pintura de contenido más grande”: 0.9 segundos
- **Total Blocking Time** “tiempo total de bloqueo”: 80 milisegundos
- **Cumulative Layout Shift** “cambio acumulativo de diseño” 0 segundos
- **Speed Index** “índice de la velocidad: 0.7 segundos

Comparativa de resultados

Pagina	Nota antes de implementar sentry	Nota después de implementar sentry
Login	99	100
Admin	91	99
Empleado	95	97
Usuario	97	99