



TEC | Tecnológico
de Costa Rica

Aseguramiento de la Calidad del Software - IC6831

Proyecto 1 Informe de Auditoría y Resultados

Profesora:
Alicia Salazar Hernandez

Integrantes:
Erika Cerdas Mejias - 2022138199
Leonardo Céspedes Tenorio - 2022080602
Kevin Chang Chang - 2022039050
Frankmin Feng Zhong - 202208924

Fecha de Entrega: 23 de Octubre

II Semestre 2024

Índice

Propósito del documento.....	3
Alcance del Sistema Bajo Auditoría.....	3
Identificación de los Elementos de Prueba.....	3
Métodos de Prueba Utilizados.....	4
Pruebas Estáticas:.....	4
4. Pruebas de Completitud Funcional.....	4
5. Pruebas de Correctitud Funcional.....	4
6. Pruebas de Mantenibilidad.....	4
7. Evaluación de Deuda Técnica.....	5
Pruebas Dinámicas:.....	5
Selenium.....	5
Lighthouse.....	6
SonarQube.....	6
Plan de pruebas.....	6
1. Autenticación.....	7
2. Gestión de Estudiantes.....	7
3. Gestión de Profesores.....	10
4. Gestión de Asistentes.....	11
5. Gestión del plan de Trabajo.....	12
6. Gestión del Equipo de Trabajo.....	13
7. Gestión de Actividades.....	13
8. Accesibilidad.....	16
9. Funcionalidad.....	16
10. Deuda Técnica.....	17
Resultados de las pruebas.....	18
Análisis de la prueba de Lighthouse.....	19
Análisis de la prueba de Funcionalidad con 3 hilos.....	30
Análisis de la deuda técnica.....	31

Propósito del documento

El propósito de este documento es detallar el proceso de auditoría de un sistema de información previamente desarrollado. La auditoría incluye la elaboración de un plan de pruebas estáticas y dinámicas, ejecución de las pruebas y la presentación de los resultados en un informe que expone los hallazgos, el análisis de la deuda técnica y las recomendaciones pertinentes.

Alcance del Sistema Bajo Auditoría

El sistema auditado es un sistema de gestión académica que permite la autenticación de usuarios, la gestión de estudiantes, profesores y asistentes, así como la generación de reportes. Este sistema fue desarrollado en un curso anterior de Diseño de Software IC-6821.

El análisis cubre las siguientes funcionalidades:

- Autenticación de usuarios.
- Registro y gestión de estudiantes.
- Registro y asignación de roles a profesores.
- Gestión de planes de trabajo y actividades.

Identificación de los Elementos de Prueba

Las pruebas se enfocaron en las siguientes funcionalidades del sistema:

- **Autenticación de Usuarios:** Validar los flujos de inicio de sesión y manejo de credenciales.
- **Gestión de Estudiantes:** Pruebas de carga de estudiantes mediante archivo CSV, visualización y filtrado.
- **Gestión de Profesores y Asistentes:** Registro, asignación de roles y gestión de actividades.
- **Generación de Reportes:** Validar la correcta generación y descarga de reportes de estudiantes por sede.

Métodos de Prueba Utilizados

Pruebas Estáticas:

4. Pruebas de Completitud Funcional

Se realizó una revisión exhaustiva del sistema para verificar que todos los requerimientos especificados están completamente implementados en el código fuente y la documentación técnica. La metodología utilizada fue el análisis de trazabilidad, donde cada requerimiento se relacionó con la funcionalidad implementada.

- **Metodología:** Se compararon los documentos de especificación de requisitos con las funciones implementadas en el sistema. Se verificó que cada requisito tuviera su correspondiente implementación en el código.
- **Resultado:** Todos los requerimientos fueron cubiertos en el sistema, sin encontrarse requisitos no implementados.

5. Pruebas de Correctitud Funcional

Mediante una revisión manual del código y pruebas unitarias disponibles, se validó que las implementaciones cumplen con las especificaciones de los requisitos asignados.

- **Metodología:** Se analizaron los métodos y funciones principales del sistema para asegurar que su lógica de programación cumple con lo esperado según los requisitos. También se utilizaron herramientas de análisis estático para identificar posibles errores lógicos y fallas en la implementación.
- **Resultado:** No se encontraron errores significativos en el código fuente y la funcionalidad está correctamente alineada con las especificaciones. Las pruebas unitarias cubren los escenarios más críticos, asegurando la correcta funcionalidad.

6. Pruebas de Mantenibilidad

Para evaluar la mantenibilidad del sistema, se realizaron análisis sobre la modularidad, la capacidad de ser modificado, y la capacidad de ser probado. Se utilizó SonarQube para analizar la estructura del código.

- **Metodología:** Se utilizó un proyecto de SonarQube y herramientas de CLI para analizar el código fuente del sistema, enfocándose en modularidad, capacidad de ser modificado y capacidad de ser probado.

- **Resultado:** Se obtuvieron índices y valores que serán analizados con detalle posteriormente en el documento.

7. Evaluación de Deuda Técnica

El análisis de deuda técnica se realizó utilizando SonarQube, donde se identificaron áreas problemáticas en cuanto a mantenibilidad y calidad del código.

- **Metodología:** A través de SonarQube, se llevó a cabo una evaluación de deuda técnica para identificar duplicación de código, complejidad ciclomática, y violaciones de calidad. El análisis incluyó la medición de la deuda técnica en términos de esfuerzo necesario para corregir problemas.
- **Resultado:** Se utilizarán los valores obtenidos con esta herramienta para hacer un análisis más profundo respecto a la deuda técnica.

Pruebas Dinámicas:

Se utilizaron herramientas como Selenium para la automatización de pruebas funcionales y Lighthouse para auditorías de accesibilidad y rendimiento. Las pruebas incluyeron:

- **Pruebas Funcionales Dinámicas:** Validación de la funcionalidad mediante pruebas de caja negra.
- **Pruebas de Rendimiento:** Simulación de múltiples usuarios para evaluar el tiempo de respuesta bajo carga.
- **Pruebas de Usabilidad:** Evaluación de la interfaz según las pautas WCAG.

Selenium

Selenium es un conjunto de herramientas de código abierto para la automatización de navegadores web. Dentro de sus características principales se encuentran:

- Soporta múltiples navegadores (Chrome, Firefox, Safari, Edge, etc.)
- Permite la escritura de pruebas en varios lenguajes de programación (Java, Python, C#, Ruby, JavaScript)
- Proporciona una API para interactuar con elementos web
- Puede ser integrado con frameworks de pruebas como JUnit, TestNG, etc.
- Soporta ejecución paralela de pruebas

Esta herramienta es comúnmente utilizada para procesos como:

- Pruebas funcionales de aplicaciones web
- Pruebas de regresión

- Automatización de tareas repetitivas en navegadores

Lighthouse

Lighthouse es una herramienta de código abierto desarrollada por Google para auditar la calidad de páginas web. Presenta características importantes como:

- Analiza el rendimiento, accesibilidad, SEO y mejores prácticas de desarrollo web
- Genera informes detallados con puntuaciones y sugerencias de mejora
- Puede ser ejecutado como una extensión de Chrome
- Soporta auditorías personalizadas

Entre sus usos comunes se encuentran:

- Optimización del rendimiento de sitios web
- Mejora de la accesibilidad
- Verificación de buenas prácticas de desarrollo web
- Auditorías de SEO

SonarQube

SonarQube es una herramienta de código abierto que se utiliza para el análisis continuo de la calidad del código fuente. Algunas de sus características principales incluyen:

- Soporta múltiples lenguajes de programación (Java, Python, C, C++, JavaScript, PHP, entre otros).
- Detecta problemas en el código como bugs, vulnerabilidades de seguridad y malos olores (code smells).
- Genera informes detallados con métricas sobre la calidad del código.
- Se integra con sistemas de control de versiones como Git, GitLab, GitHub, Bitbucket.
- Proporciona reglas de calidad y seguridad personalizables para distintos lenguajes de programación.
- Permite la integración con pipelines de CI/CD como Jenkins, GitLab CI, Travis CI, etc.

Entre los usos comunes de SonarQube se encuentran:

- Mejora de la calidad del código a través de la identificación de errores y malos olores.
- Prevención de vulnerabilidades de seguridad en el código fuente.
- Evaluación de la deuda técnica para facilitar la refactorización.
- Control continuo de calidad en proyectos de desarrollo de software.
- Automatización de revisiones de código en equipos de desarrollo.

Plan de pruebas

1. Autenticación

ID	IT001 Inicio de Sesión
Objetivo	Verificar que el usuario puede iniciar sesión
Descripción	Este caso de prueba valida el flujo de iniciar sesión en el sistema.
Precondiciones	El usuario debe de tener una cuenta válida registrada en la base de datos.
Pasos	<ul style="list-style-type: none">• Navegar a la página de inicio del sistema.• Hacer click en el botón de "Iniciar sesión".• Ingresar credenciales inválidas (usuario y/o contraseña incorrecta).• Verificar que el sistema muestre un mensaje de error.• Ingresar credenciales válidas (usuario y contraseña correctos).• Verificar que el sistema permite acceder correctamente a la página
Condiciones	Solo se debe haber iniciado sesión correctamente con credenciales válidas.
Métricas	Tiempo de respuesta del sistema desde que se ingresan las credenciales hasta que se otorga o deniega el acceso. Número de intentos fallidos antes de iniciar sesión exitosamente. Mensajes de error mostrados en caso de credenciales incorrectas
Estado	

2. Gestión de Estudiantes

ID	IT002 Registro de Estudiantes
Objetivo	Verificar el registro exitoso de un estudiante mediante un archivo CSV
Descripción	La prueba valida que se pueda cargar un archivo CSV y registrar a un o varios estudiantes en el sistema de manera correcta.
Precondiciones	El archivo CSV con los datos del estudiante debe existir.
Pasos	<ul style="list-style-type: none">• Acceder al sitio web y navegar a la página de inicio de sesión.• Ingresar las credenciales del administrador• Hacer click en "Iniciar Sesión".• Navegar a "Estudiantes"• Hacer click en "Registrar Estudiante".• Seleccionar el archivo CSV.• Hacer click en "Registrar".
Condiciones	El archivo CSV debe ser cargado correctamente
Métricas	Tiempo de respuesta del sistema sobre el procesamiento del CSV

Estado	
--------	--

ID	IT003 Visualización de Estudiantes
Objetivo	Verificar la visualización de la lista de estudiantes registrados
Descripción	Validar que la lista de estudiantes sea accesible y se pueda navegar correctamente a la sección de estudiantes
Precondiciones	Deben existir estudiantes registrados
Pasos	<ul style="list-style-type: none"> ● Acceder al sitio web y navegar a la página de inicio de sesión. ● Ingresar las credenciales del administrador ● Hacer click en "Iniciar Sesión". ● Navegar a "Estudiantes" ● Verificar que se despliega correctamente la lista de estudiantes
Condiciones	La lista de estudiantes debe ser visible
Métricas	Tiempo de respuesta sobre la carga de la lista estudiantes
Estado	

ID	IT004 Filtro por Apellido
Objetivo	Verificar la funcionalidad de filtro de estudiantes por apellido
Descripción	Se busca validar que los estudiantes se ordenen correctamente por apellido al aplicar el filtro
Precondiciones	Deben existir estudiantes registrados
Pasos	<ul style="list-style-type: none"> ● Acceder al sitio web y navegar a la página de inicio de sesión. ● Ingresar las credenciales del administrador ● Hacer click en "Iniciar Sesión". ● Navegar a "Estudiantes" ● Aplicar el filtro por apellido ● Verificar que se despliega correctamente la lista de estudiantes
Condiciones	La lista de estudiantes debe ser visible y ordenada por Apellido
Métricas	Tiempo de respuesta sobre la carga de la lista estudiantes
Estado	

ID	IT005 Filtro por Carnet
Objetivo	Verificar la funcionalidad de filtro de estudiantes por carnet
Descripción	Se busca validar que los estudiantes se ordenen correctamente por carnet al aplicar el filtro
Precondiciones	Deben existir estudiantes registrados
Pasos	<ul style="list-style-type: none"> ● Acceder al sitio web y navegar a la página de inicio de sesión. ● Ingresar las credenciales del administrador ● Hacer click en "Iniciar Sesión". ● Navegar a "Estudiantes" ● Aplicar el filtro por carnet ● Verificar que se despliega correctamente la lista de estudiantes
Condiciones	La lista de estudiantes debe ser visible y ordenada por carnet
Métricas	Tiempo de respuesta sobre la carga de la lista estudiantes
Estado	

ID	IT006 Generación de Reporte
Objetivo	El usuario puede generar un reporte con todos los estudiantes divididos por sede.
Descripción	Se verifica si el usuario es capaz de entrar a la pestaña de estudiantes creando un reporte en excel con todos los estudiantes.
Precondiciones	Debe de iniciar sesión. Espacio en la computadora para descargar el archivo.
Pasos	<ul style="list-style-type: none"> ● Navegar a la página de inicio. ● Hacer click en el botón de inicio de sesión. ● Ingresar las credenciales válidas. ● Hacer click en el enlace "Estudiantes". ● Hacer click en el enlace de "Generar reporte" ● Escoger la opción de "Todos los Campus" ● Finalmente seleccionar el botón de "submit"
Condiciones	Se descarga el reporte en formato .xlsx y el reporte contiene todos los estudiantes existentes divididos por campus.
Métricas	La descarga automática del reporte. La cantidad de fallos en la descarga.
Estado	

3. Gestión de Profesores

ID	IT007 Registrar Profesor
Objetivo	Verificar que el sistema registra correctamente a los profesores en la BD.
Descripción	Valida el flujo completo de registro de profesores, asegurando que los datos se almacenen correctamente en la BD y que se cumplan las restricciones.
Precondiciones	El usuario debe tener permisos administrativos o de gestión.
Pasos	<ul style="list-style-type: none"> ● Navegar a la página de inicio. ● Hacer click en el botón de inicio de sesión. ● Ingresar las credenciales válidas. ● Hacer click en el enlace "Profesores". ● Seleccionar la opción de "Registrar Profesor". ● Rellenar los campos solicitados ● Hacer click en el botón de "Registrar"
Condiciones	El registro de profesores debe realizarse correctamente solo cuando todos los datos ingresados sean válidos. El sistema debe mostrar mensajes de error adecuados en caso de errores en el ingreso de datos.
Métricas	Tiempo de respuesta del sistema desde que se hace click en "Registrar" hasta que se confirma el registro.
Estado	

ID	IT008 Asignar Coordinador
Objetivo	Verificar que se le puede asignar el rol de coordinador
Descripción	Este caso de prueba valida el flujo completo de poder asignar el rol de coordinador a un profesor.
Precondiciones	Deben haber equipo de trabajo y profesores existentes registrados en la base de datos.
Pasos	<ul style="list-style-type: none"> ● Navegar a la página de inicio. ● Hacer click en el botón de inicio de sesión. ● Ingresar las credenciales válidas. ● Hacer click en el enlace "Profesores". ● Seleccionar la opción de "Dar Rol Coordinador". ● Seleccionar el equipo de trabajo deseado ● Seleccionar el profesor al cual se le desea dar el rol de coordinador.
Condiciones	En la lista de profesores se debe ver reflejado el cambio
Métricas	Tiempo de respuesta para asignar el rol de coordinador.
Estado	

4. Gestión de Asistentes

ID	IT009 Registro Asistente
Objetivo	Verificar el correcto funcionamiento del registro de un asistente.
Descripción	La prueba se centra en realizar el flujo completo de registro de un asistente.
Precondiciones	El usuario debe ser administrador para poder realizar la acción
Pasos	<ul style="list-style-type: none"> ● Navegar a la página de inicio ● Hacer click en "Iniciar Sesión". ● Ingresar las credenciales del administrador (válidas) ● Hacer click en el botón "Iniciar Sesión". ● Navegar a "Equipo Guía" desde el menú principal. ● Hacer click en "Registrar Asistente". ● Completar el formulario de registro con datos válidos: ● Hacer click en el botón "Registrar".
Condiciones	<ol style="list-style-type: none"> 1. El asistente debe ser creado exitosamente. 2. El sistema debe redirigir a la URL correcta 3. No deben presentarse errores de validación ni del sistema.
Métricas	Tiempo de respuesta para crear el asistente.
Estado	

ID	IT010 Asignar Rol Asistente
Objetivo	El usuario puede asignar a algún asistente el rol de administrador y escoger a qué campus pertenece.
Descripción	Se valida si el usuario es capaz de asignar al asistente el rol administrativo.
Precondiciones	Deben de existir asistentes ya registrados en el sistema. Se debe escoger un campus.
Pasos	<ul style="list-style-type: none"> ● Navegar a la página de inicio. ● Hacer click en el botón de inicio de sesión. ● Ingresar las credenciales válidas. ● Hacer click en el enlace 'Equipo Guía'. ● Hacer click en 'Asignar Asistente Administrativo por Campus'. ● Se debe seleccionar una de las opciones de "Asistentes Disponibles". ● Se debe de seleccionar una de las opciones en el campo de "Campus" ● Cuando ambos campos estén llenos, presionar el botón de "Asignar"
Condiciones	Se tuvo que haber asignado el rol al asistente escogido en la sede escogida.
Métricas	Tiempo de respuesta para asignar el asistente.
Estado	

5. Gestión del plan de Trabajo

ID	IT011 Crear Plan
Objetivo	Verificar que un usuario pueda crear un plan exitosamente.
Descripción	Este caso de prueba valida el flujo de creación de un nuevo plan en el sistema.
Precondiciones	El usuario debe estar autenticado y tener acceso a la gestión de planes.
Pasos	<ul style="list-style-type: none">● Navegar a la página de inicio.● Hacer click en el botón de inicio de sesión.● Ingresar las credenciales válidas.● Hacer click en el enlace "Planes".● Hacer click en el botón de crear plan.● Completar todos los campos del formulario de creación de plan con datos válidos.● Hacer click en el botón de guardar plan.
Condiciones	El usuario debe ser redirigido a la página de planes y ver el nuevo plan en la lista.
Métricas	Tiempo de respuesta para crear el plan, tasa de éxito de creación de plan.
Estado	

ID	IT012 Eliminar Plan
Objetivo	Verificar que un usuario pueda eliminar un plan exitosamente.
Descripción	Este caso de prueba valida el flujo de eliminación de un plan existente en el sistema.
Precondiciones	El usuario debe estar autenticado y tener al menos un plan creado.
Pasos	<ul style="list-style-type: none">● Navegar a la página de inicio.● Hacer click en el botón de inicio de sesión.● Ingresar las credenciales válidas.● Hacer click en el enlace "Planes".● Hacer click en el botón de eliminar junto al plan que se desea eliminar.● Confirmar la eliminación del plan.
Condiciones	El plan debe ser eliminado y no debe aparecer en la lista de planes.
Métricas	Tiempo de respuesta para eliminar el plan, tasa de éxito de eliminación de plan.
Estado	

6. *Gestión del Equipo de Trabajo*

ID	IT013 Crear y Ver Equipo de Trabajo
Objetivo	Verificar que un usuario pueda generar y ver un equipo de trabajo
Descripción	Se valida si el usuario puede crear equipo de trabajo conformado con 5 profesores de diferentes sedes. También se puede ver los equipos creados.
Precondiciones	Deben de existir 5 profesores, cada uno de una sede diferente. El usuario debe de estar registrado y tener permisos de administrador.
Pasos	<ul style="list-style-type: none">● Navegar a la página de inicio.● Hacer click en el botón de inicio de sesión.● Ingresar las credenciales válidas.● Hacer click en el enlace 'Equipo Guia'.● Seleccionar "Crear Equipo de Trabajo"● Seleccionar 5 profesores de diferentes campus.● Escoger la generación a cargo.● Finalmente se debe presionar el botón de crear.
Condiciones	El usuario al entrar a la vista de equipos debe de poder ver el equipo creado.
Métricas	El equipo se crea con los profesores escogidos. Se toman en cuenta las restricciones que no permite tener menos de 5 profesores de la misma sede.
Estado	

7. *Gestión de Actividades*

ID	IT014 Crear Actividad
Objetivo	Verificar que un usuario pueda crear una actividad exitosamente.
Descripción	Este caso valida el flujo de creación de una nueva actividad en el sistema.
Precondiciones	El usuario debe estar autenticado y tener acceso a la gestión de actividades.
Pasos	<ul style="list-style-type: none">● Navegar a la página de inicio.● Hacer click en el botón de inicio de sesión.● Hacer click en el enlace "Planes".● Seleccionar un plan existente.● Hacer click en el botón para agregar actividad.● Completar los campos del formulario de actividad con datos válidos.● Subir el archivo del afiche.● Hacer click en el botón de guardar actividad.
Condiciones	El usuario debe ser redirigido a la página del plan y ver la actividad en lista.
Métricas	Tiempo de respuesta para crear la actividad.
Estado	

ID	IT015 Editar Actividad
Objetivo	Verificar que un usuario pueda editar una actividad existente.
Descripción	Este caso de prueba valida el flujo de edición de una actividad en el sistema.
Precondiciones	El usuario debe estar autenticado y tener acceso a la gestión de actividades en un plan existente. Debe existir al menos una actividad que pueda ser editada.
Pasos	<ul style="list-style-type: none"> ● Navegar a la página de inicio. ● Hacer click en el botón de inicio de sesión. ● Hacer click en el enlace "Planes". ● Seleccionar un plan existente. ● Ver la actividad que se desea editar. ● Hacer click en el botón de editar actividad. ● Modificar el nombre de la actividad. ● Hacer click en el botón de guardar actividad.
Condiciones	El usuario debe ser redirigido a la página de la actividad y ver los cambios.
Métricas	Tiempo de respuesta para editar la actividad.
Estado	

ID	IT016 Cancelar Actividad
Objetivo	Verificar que un usuario pueda cancelar una actividad existente.
Descripción	Este caso valida el flujo de cancelación de una actividad en el sistema.
Precondiciones	El usuario debe estar autenticado y tener acceso a la gestión de actividades en un plan existente. Debe existir al menos una actividad que pueda ser cancelada.
Pasos	<ul style="list-style-type: none"> ● Navegar a la página de inicio. ● Hacer click en el botón de inicio de sesión. ● Hacer click en el enlace "Planes". ● Seleccionar un plan existente. ● Ver la actividad que se desea cancelar. ● Hacer click en el botón de editar actividad. ● Cambiar el estado de la actividad a cancelada. ● Introducir la justificación de la cancelación. ● Hacer click en el botón de guardar actividad.
Condiciones	El usuario debe ser redirigido a la página de la actividad y ver el estado.
Métricas	Tiempo de respuesta para cancelar la actividad.
Estado	

ID	IT017 Comentar Actividad
Objetivo	Verificar que un usuario pueda comentar una actividad.
Descripción	Este caso valida el flujo de agregar un comentario a una actividad.
Precondiciones	El usuario debe estar autenticado y tener acceso a la actividad en un plan.
Pasos	<ul style="list-style-type: none"> ● Navegar a la página de inicio. ● Hacer click en el botón de inicio de sesión. ● Hacer click en el enlace "Planes". ● Seleccionar un plan existente. ● Ver la actividad. ● Hacer click en el botón para comentar. ● Introducir un comentario en el campo correspondiente. ● Hacer click en el botón de guardar comentario.
Condiciones	El usuario debe ser redirigido a la página de la actividad y ver el nuevo comentario en la lista de comentarios.
Métricas	Tiempo de respuesta para agregar el comentario.
Estado	

ID	IT018 Comentar sobre comentario
Objetivo	Verificar que un usuario pueda comentar sobre un comentario existente.
Descripción	Valida el flujo de agregar un comentario a un comentario existente.
Precondiciones	El usuario debe estar autenticado y tener acceso a la actividad en un plan existente. Debe existir al menos un comentario al que se pueda responder.
Pasos	<ul style="list-style-type: none"> ● Navegar a la página de inicio. ● Hacer click en el botón de inicio de sesión. ● Hacer click en el enlace "Planes". ● Seleccionar un plan existente. ● Ver la actividad. ● Hacer click en el botón para comentar sobre un comentario existente. ● Introducir un comentario en el campo correspondiente. ● Hacer click en el botón de guardar comentario
Condiciones	El usuario debe ser redirigido a la página de la actividad y ver el nuevo comentario en la lista de comentarios.
Métricas	Tiempo de respuesta para agregar el comentario sobre otro comentario.
Estado	

8. Accesibilidad

ID	IT019 Accesibilidad de la Página
Objetivo	Verificar que la aplicación cumpla con los estándares de accesibilidad según las pautas WCAG.
Descripción	Este caso de prueba valida la accesibilidad de la aplicación web utilizando Lighthouse.
Precondiciones	La aplicación web debe estar en línea y accesible. El usuario debe tener instalada la extensión Lighthouse en su navegador Chrome.
Pasos	<ul style="list-style-type: none">● Abrir Google Chrome.● Navegar a la URL de la aplicación web (https://fonmala.nyc.dom.my.id/).● Hacer click en el ícono de Lighthouse en la barra de herramientas del navegador.● Seleccionar "Accessibility" en las opciones de auditoría.● Hacer click en "Generate report" (Generar informe).● Esperar a que Lighthouse complete la auditoría.
Condiciones	El informe de Lighthouse debe completarse sin errores que indiquen problemas de accesibilidad.
Métricas	Puntuación de accesibilidad (de 0 a 100), lista de problemas de accesibilidad encontrados.
Estado	

9. Funcionalidad

ID	IT020 Funcionalidad de la Página
Objetivo	Verificar la correcta ejecución concurrente de múltiples instancias.
Descripción	Valida que el script permite ejecutar varios hilos que ejecutan el archivo especificado al mismo tiempo, comprobando que se ejecuten en paralelo.
Precondiciones	Python debe estar instalado. El script debe estar accesible. El archivo que se va a ejecutar debe estar presente y ser ejecutable.
Pasos	<ul style="list-style-type: none">● Acceder a la ruta del proyecto● Ejecutar el comando en la terminal para probar con 3 hilos● Ver resultados en terminal
Condiciones	El script debe iniciar la ejecución del file en 3 hilos concurrentes. Cada hilo debe imprimir el mensaje "Starting file execution..." y "File execution completed." en la terminal.
Métricas	Tiempo de ejecución total del script. Número de hilos iniciados
Estado	

10. Deuda Técnica

ID	IT021 Deuda Técnica
Objetivo	Realizar una evaluación que identifique problemas de mantenimiento, código duplicado, complejidad del código y otros factores relacionados con la deuda técnica.
Descripción	Este caso de prueba utiliza SonarQube para analizar el código fuente y detectar problemas que contribuyan a la deuda técnica, como duplicación de código, complejidad ciclomática, violaciones de patrones de codificación y otros factores de calidad.
Precondiciones	El proyecto debe estar configurado y disponible en el repositorio de código. SonarQube debe estar instalado y configurado con acceso al proyecto.
Pasos	<ul style="list-style-type: none">● Abrir SonarQube en el navegador y acceder con las credenciales correspondientes.● Enlazar el proyecto de código fuente a la instancia de SonarQube.● Ejecutar el análisis de SonarQube mediante el comando de CLI.● Esperar a que finalice el análisis.● Revisar el informe generado en SonarQube.
Condiciones	El análisis de SonarQube debe completarse sin errores y generar un informe detallado con métricas sobre complejidad, duplicación de código, violaciones de patrones de codificación y otros factores que contribuyen a la deuda técnica.
Métricas	Índice de Deuda Técnica (Technical Debt Ratio), Porcentaje de código duplicado, Complejidad ciclomática promedio por función, Violaciones de patrones de codificación, Tamaño promedio por función
Estado	

Resultados de las pruebas

ID	Tiempo de Ejecución	Resultado	Porcentaje de Éxito	Observaciones
TC001	28.604s	Exitosa	100%	Se hacen 6 pruebas, con credenciales válidas e inválidas
TC002	27.756s	Exitosa	100%	Se hicieron 2 pruebas una correcta y una que debe de dar error
TC003	6.173s	Exitosa	100%	Sin comentarios
TC004	7.566s	Exitosa	100%	Sin comentarios
TC005	6.869s	Exitosa	100%	Sin comentarios
TC006	12.836s	Exitosa	100%	Sin Comentarios
TC007	7.630s	Exitosa	100%	Se despliega información relevante en la terminal
TC008	8.383s	Exitosa	100%	Sin comentarios
TC009	22.183s	Exitosa	100%	Se hicieron 2 pruebas una correcta y una que debe de dar error
TC010	12.801s	Exitosa	100%	Sin comentarios
TC011	8.102s	Exitosa	100%	Sin comentarios
TC012	8.297s	Exitosa	100%	Sin comentarios
TC013	13.574s	Exitosa	100%	Sin comentarios
TC014	9.764s	Exitosa	100%	Sin comentarios
TC015	7.241s	Exitosa	100%	Sin comentarios
TC016	7.006s	Exitosa	100%	Sin comentarios
TC017	7.785s	Exitosa	100%	Sin comentarios
TC018	7.111s	Exitosa	100%	Sin comentarios

Observaciones: Todas las pruebas se completaron con éxito. No se encontraron errores críticos en las funcionalidades probadas.

Análisis de la prueba de Lighthouse

La prueba de accesibilidad nos permite evaluar que tan utilizable puede ser un producto (usualmente de software) para personas que sufren de alguna forma de discapacidad. Estas discapacidades pueden ser: discapacidad visual, auditiva, motora o cognitiva. Al haber realizado la prueba de accesibilidad, se puede ver que todas las páginas con excepción de la página de inicio y la página de creación del equipo lograron sacar más de 90 en el score de accesibilidad.

La página de inicio no logró pasar la prueba debido a que los colores del texto tienen un bajo contraste, lo que hace que se mezclen demasiado con el fondo, dificultando o incluso imposibilitando que muchos usuarios puedan leer los textos con comodidad. Este tipo de errores puede tener mayor impacto en las personas con discapacidades visuales, personas con daltonismo o adultos mayores que tienen menor sensibilidad a los cambios de contraste.

Para solucionar este problema, se recomienda usar colores que contrastan adecuadamente entre el fondo y el texto, evitando mezclar textos oscuros con fondos oscuros o textos claros con fondos claros. Otras consideraciones incluyen ajustar la saturación y el brillo, de manera que se pueda aumentar la accesibilidad de la página mientras se conservan los colores originales.

La otra página que no logró superar el score de 90 en la prueba de accesibilidad fue la página de creación del equipo. Esta página sufre del mismo problema de contraste donde existe un combo-box que utilice dos tonos de grises para desplegar los datos de años.

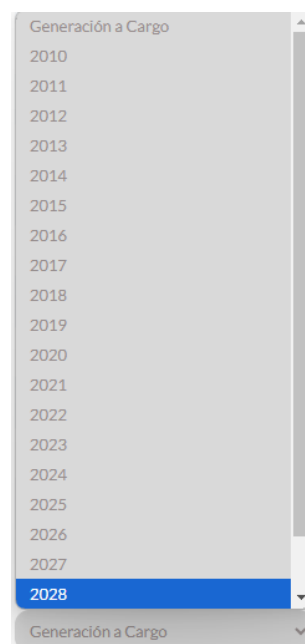


Figura #1 Combo box con carencias de accesibilidad

Además, se indica que otro factor que dificulta la accesibilidad de la página es la falta de etiquetas para los elementos del formulario. La ausencia de estas etiquetas no solo afecta a las personas con discapacidad, sino que también puede representar un desafío para quienes no tienen ninguna discapacidad, ya que su falta puede generar confusión y omitir el contexto necesario para entender lo que se solicita.

A continuación se presentan las imágenes correspondientes a las pruebas de accesibilidad para cada una de las pestañas en la página web:

1.1 Lighthouse prueba de accesibilidad del homepage en la diapositiva 2 falta sonarquebe

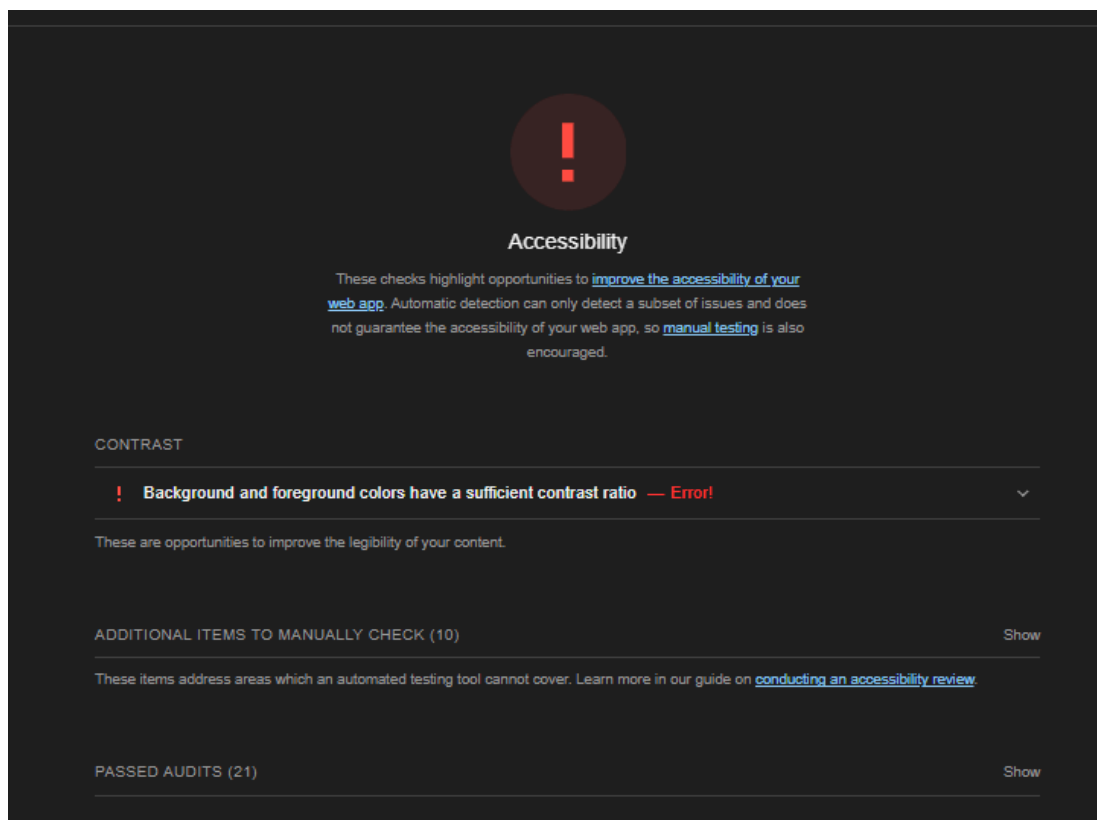


Figura #2 Nota de accesibilidad de la página inicial

1.2 Lighthouse prueba de accesibilidad de la página de asignar asistente

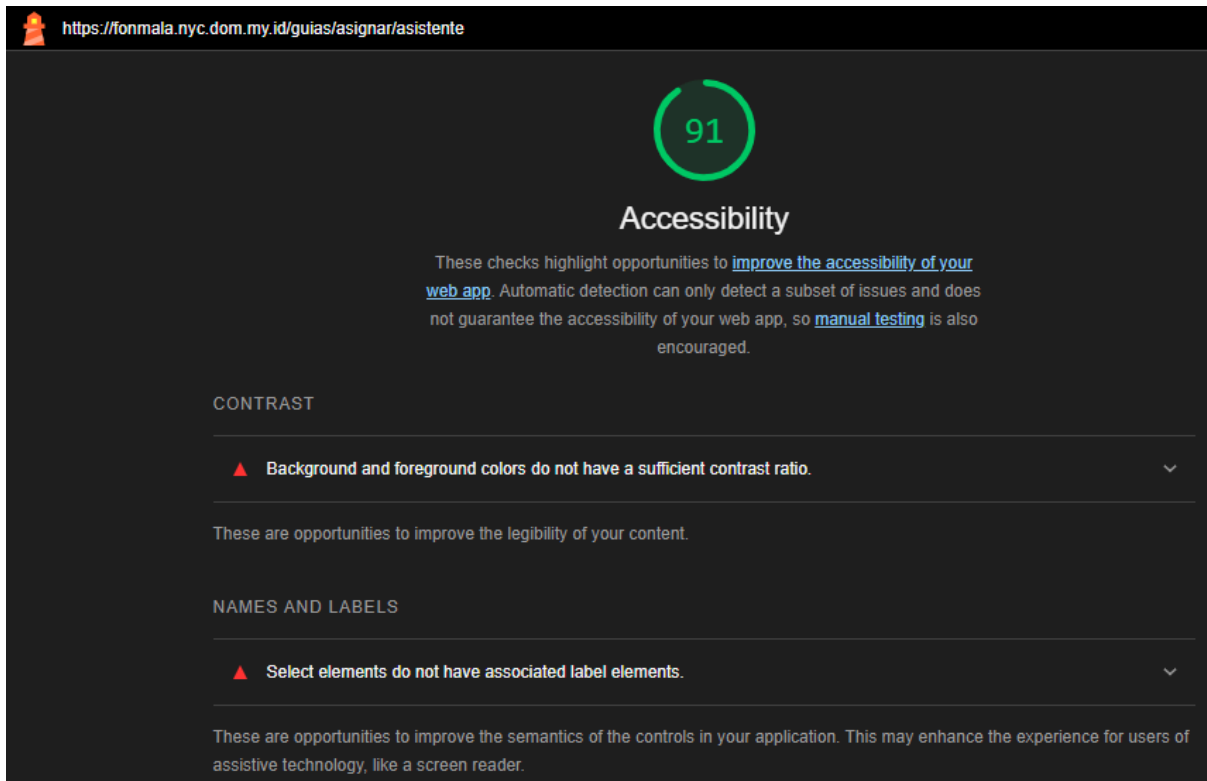


Figura #3 Nota de accesibilidad de la página de asignación de asistente

1.3 Lighthouse prueba de accesibilidad de la página de registrar asistente

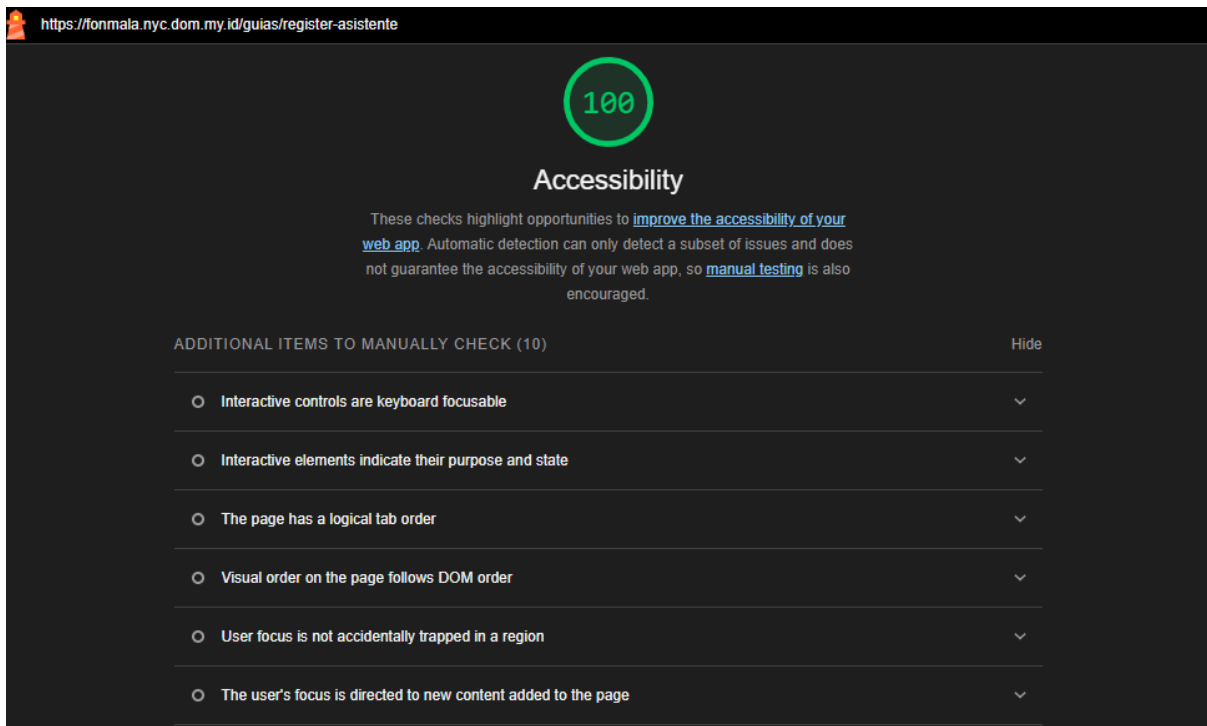


Figura #3 Nota de accesibilidad de la página para registrar asistentes

1.4 Lighthouse prueba de accesibilidad de la página de crear equipos

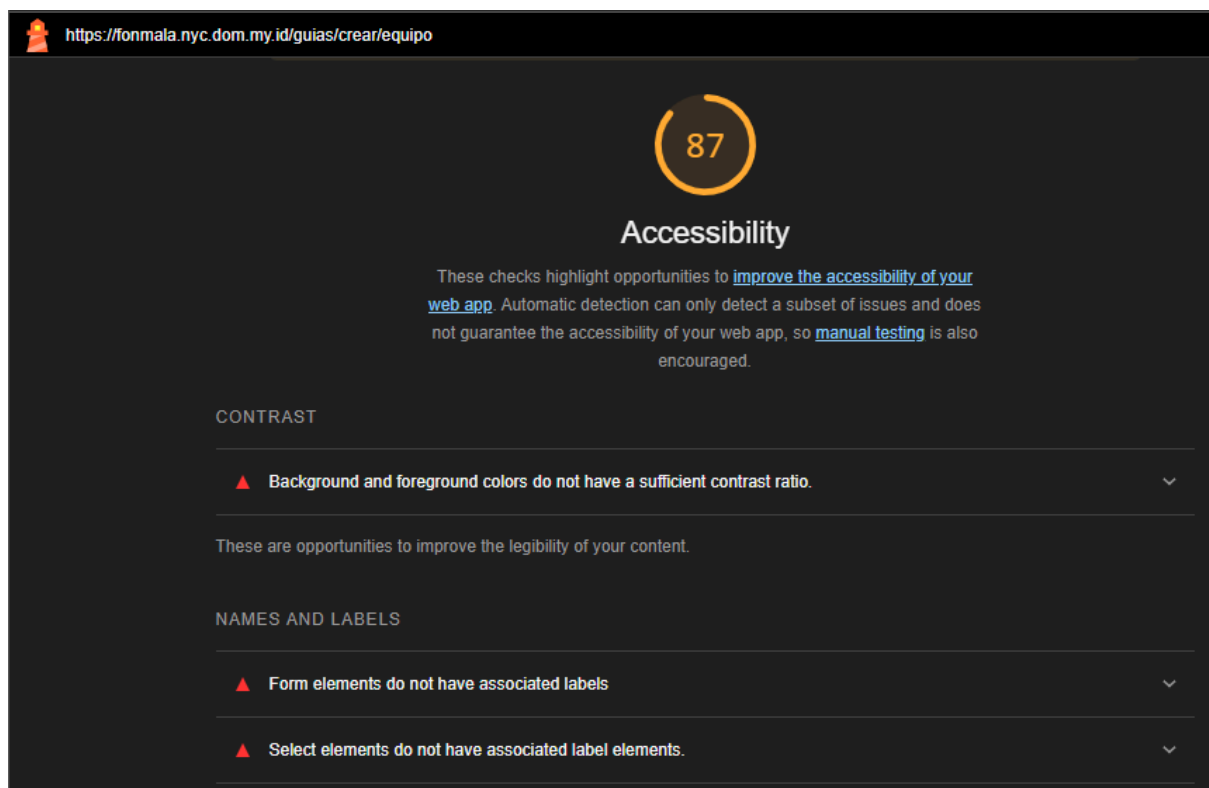


Figura #4 Nota de accesibilidad de la página para crear equipos

1.5 Lighthouse prueba de accesibilidad de la página de eliminar equipos

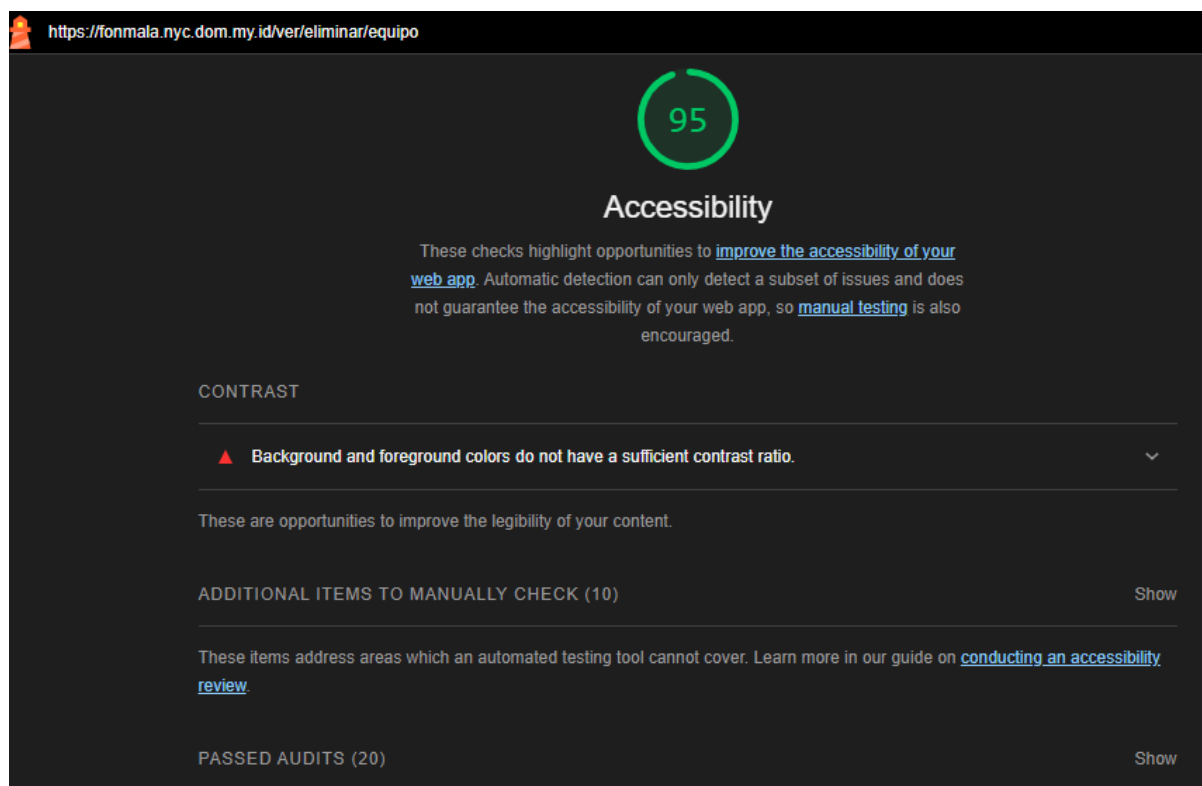


Figura #5 Nota de accesibilidad de la página para eliminar equipos

1.6 Lighthouse prueba de accesibilidad de la página de registrar profesores

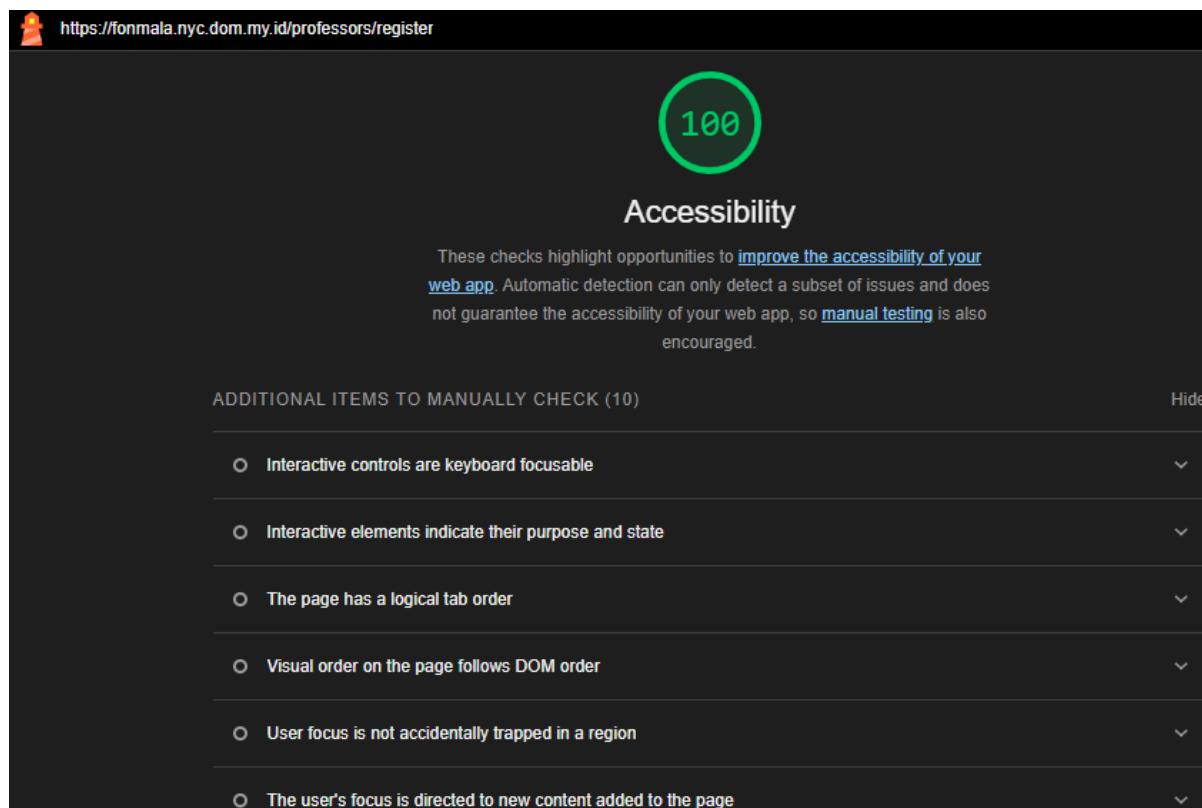


Figura #6 Nota de accesibilidad de la página para registrar profesores

1.7 Lighthouse prueba de accesibilidad de la página de asignar rol de profesor coordinador

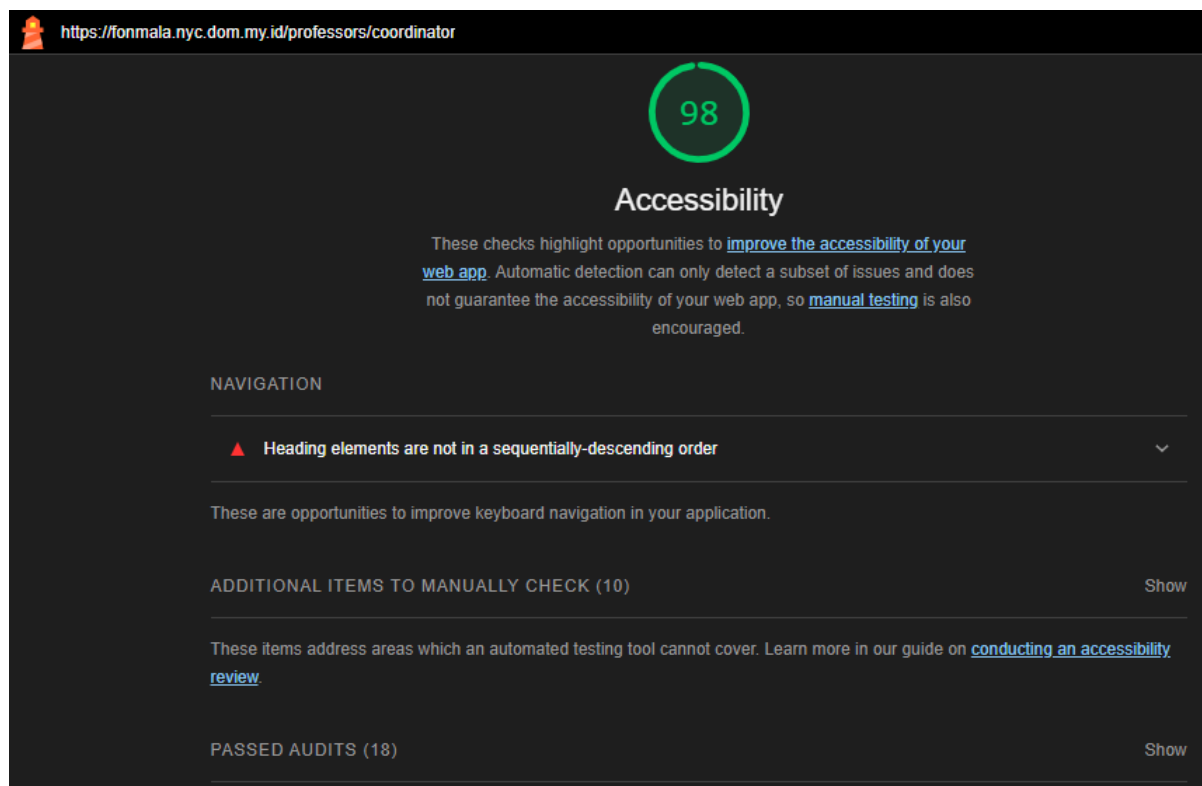


Figura #7 Nota de accesibilidad de la página para asignar rol de coordinador

1.8 Lighthouse prueba de accesibilidad de la página para registrar estudiantes

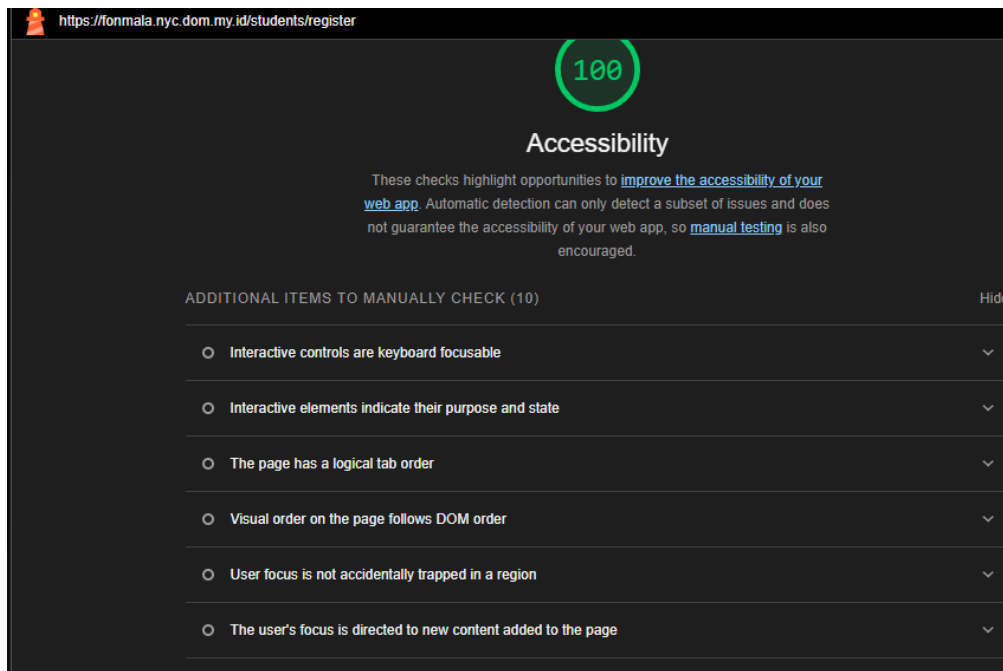


Figura #8 Nota de accesibilidad de la página para registrar estudiantes

1.9 Lighthouse prueba de accesibilidad de la página para generar reportes

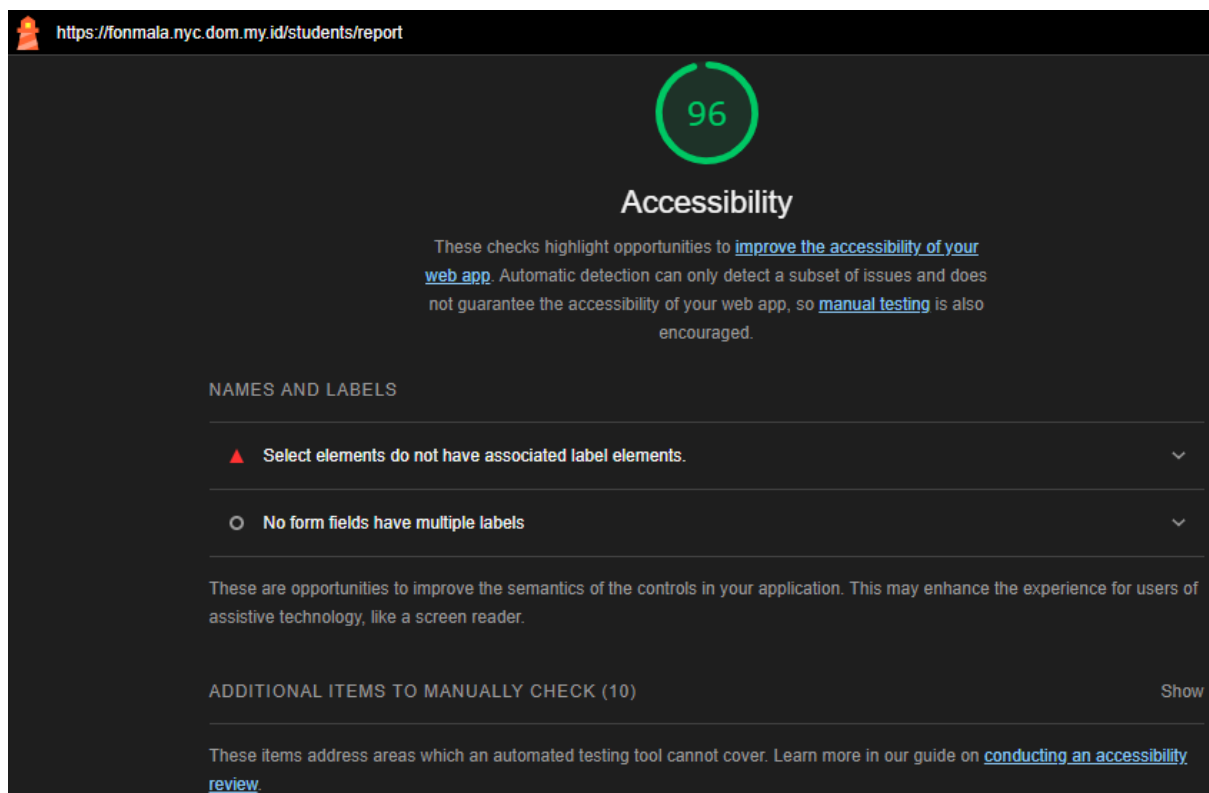


Figura #9 Nota de accesibilidad de la página para generar reportes

1.10 Lighthouse prueba de accesibilidad de la página para ver estudiantes

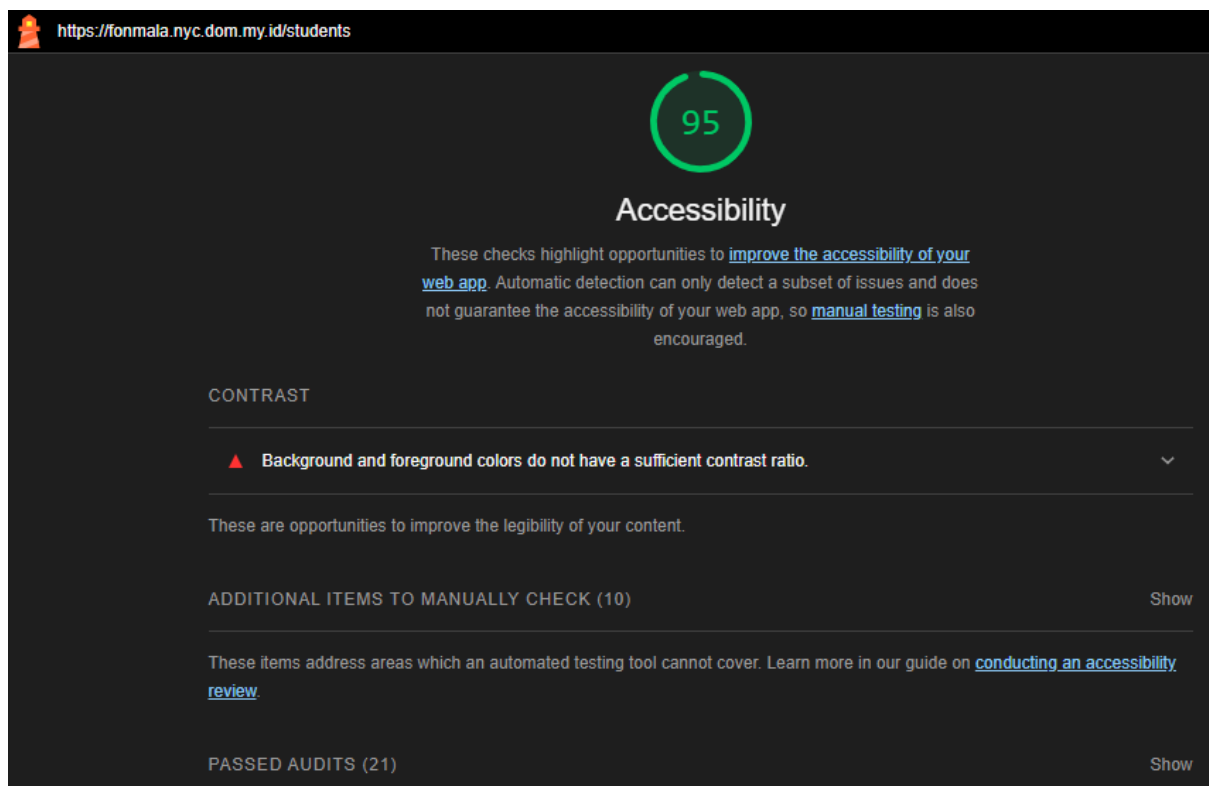


Figura #10 Nota de accesibilidad de la página para ver estudiantes

1.11 Lighthouse prueba de accesibilidad de la página para ver profesores

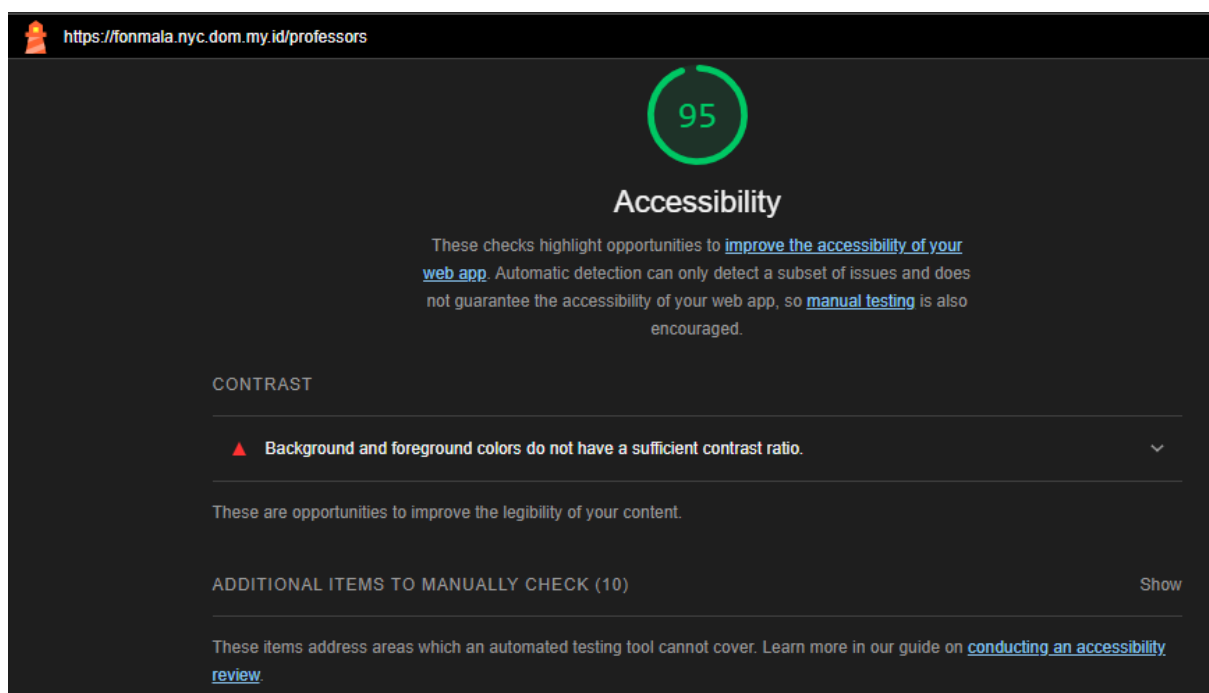


Figura #11 Nota de accesibilidad de la página para ver profesores

1.12 Lighthouse prueba de accesibilidad de la página para ver planes

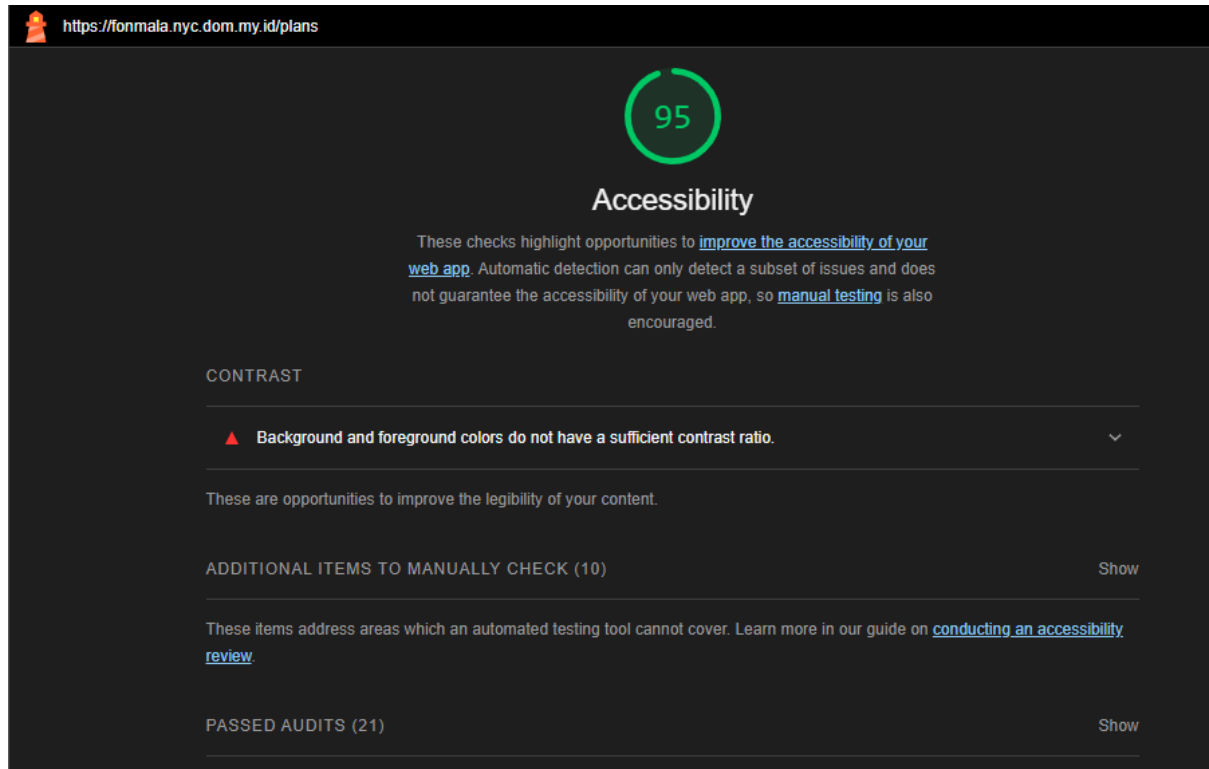


Figura #12 Nota de accesibilidad de la página para ver planes

1.13 Lighthouse prueba de accesibilidad de la página para ver guías.

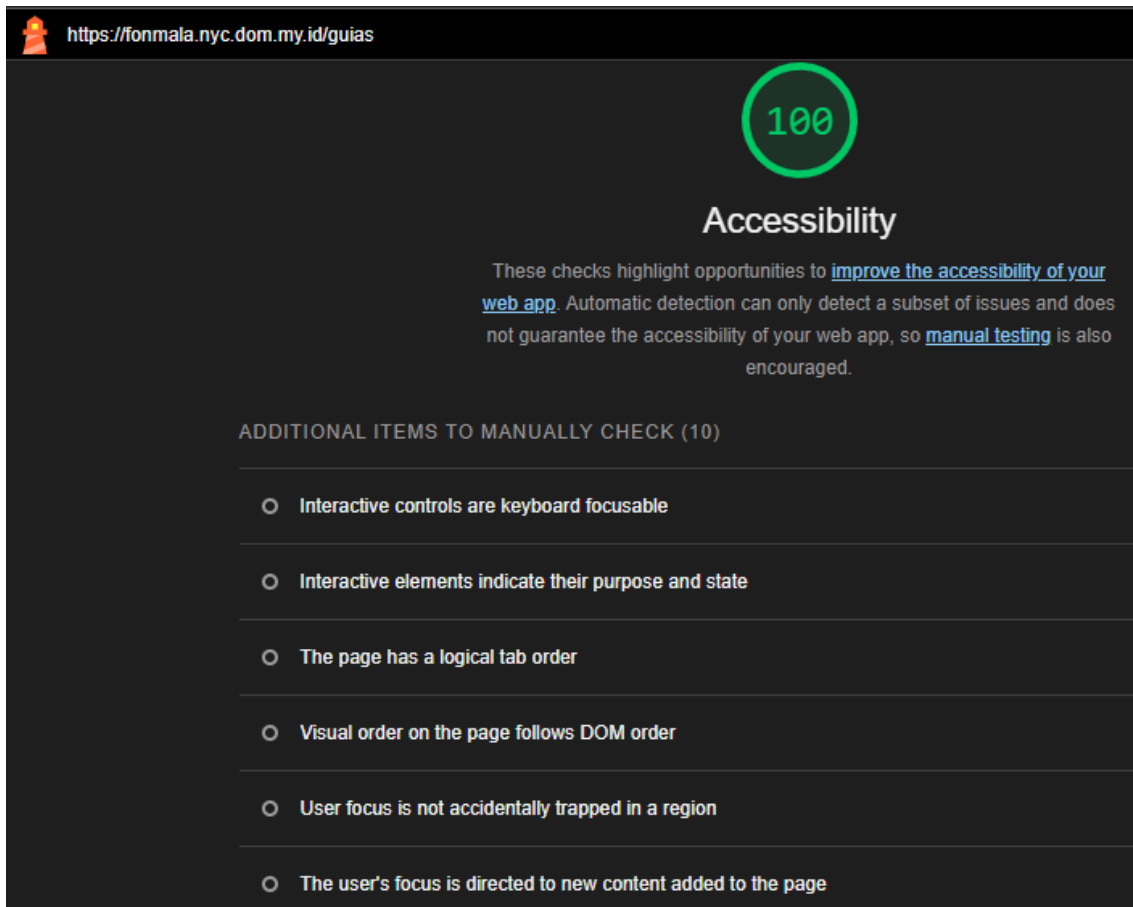


Figura #13 Nota de accesibilidad de la página para ver el menú de guías

1.14 Lighthouse prueba de accesibilidad de la página para crear planes.

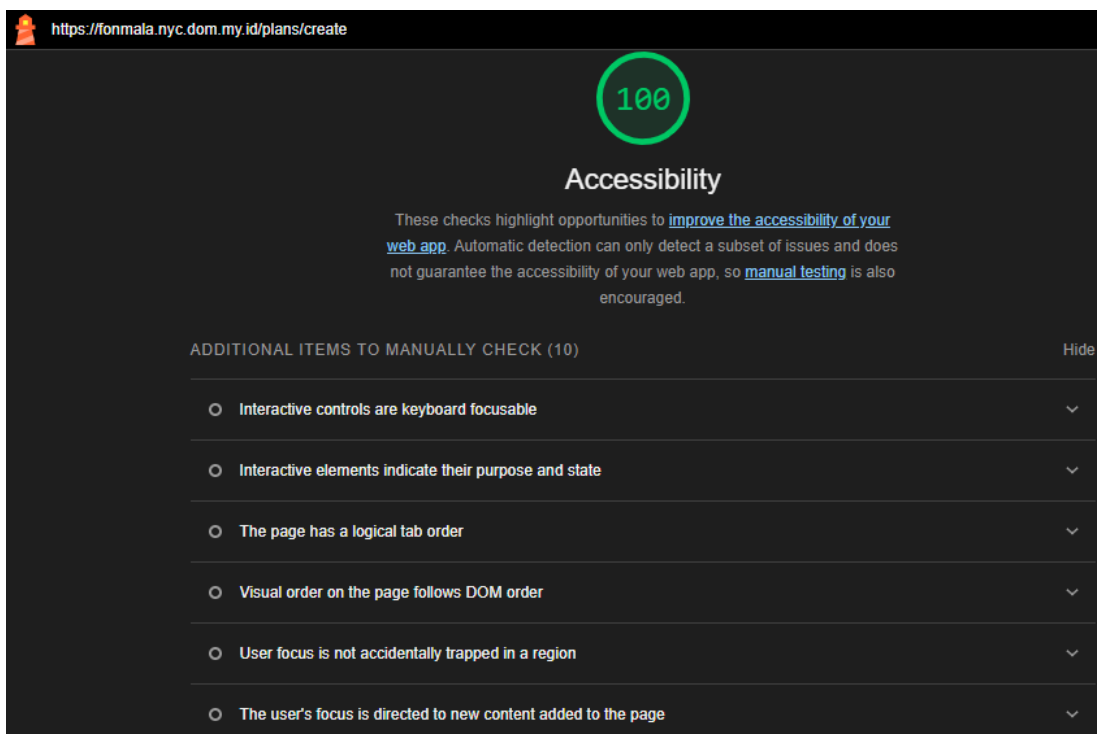


Figura #14 Nota de accesibilidad de la página para crear planes

1.15 Lighthouse prueba de accesibilidad de la página para ver la actividad.

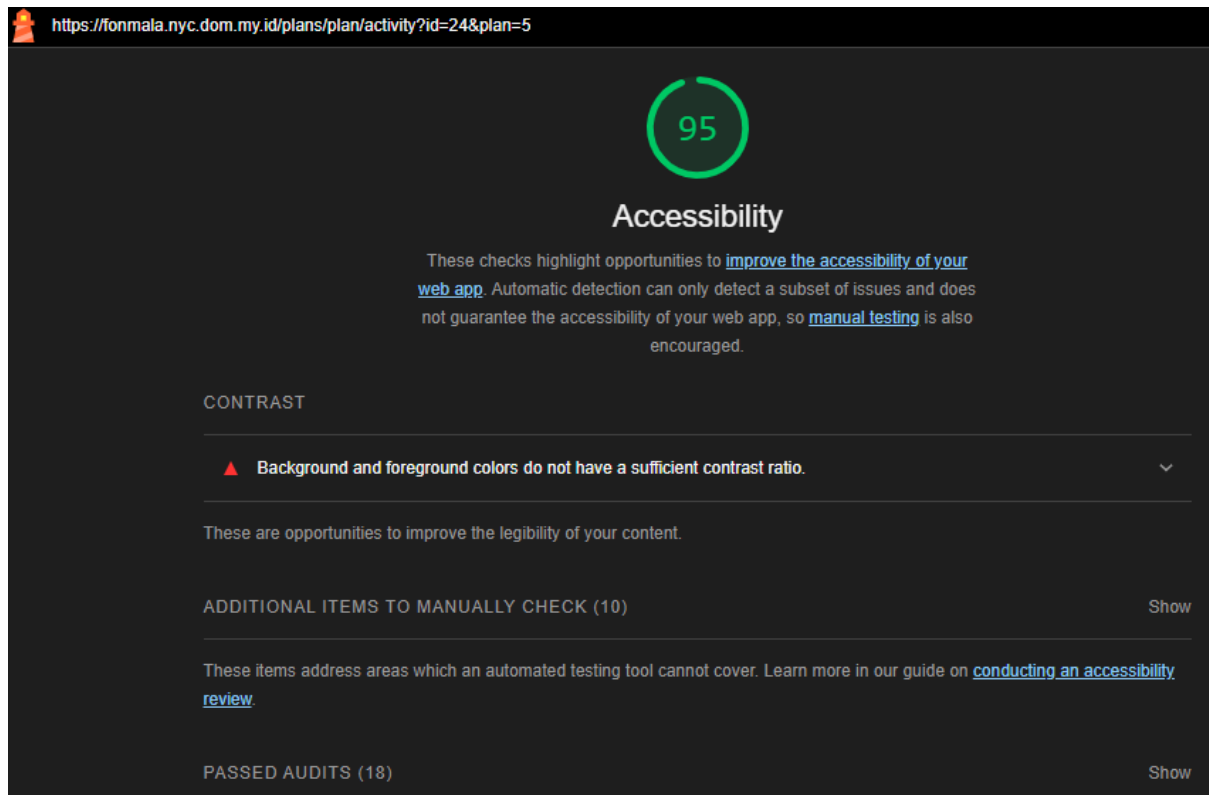


Figura #15 Nota de accesibilidad de la página para ver actividades

1.16 Lighthouse prueba de accesibilidad de la página para ver el usuario .

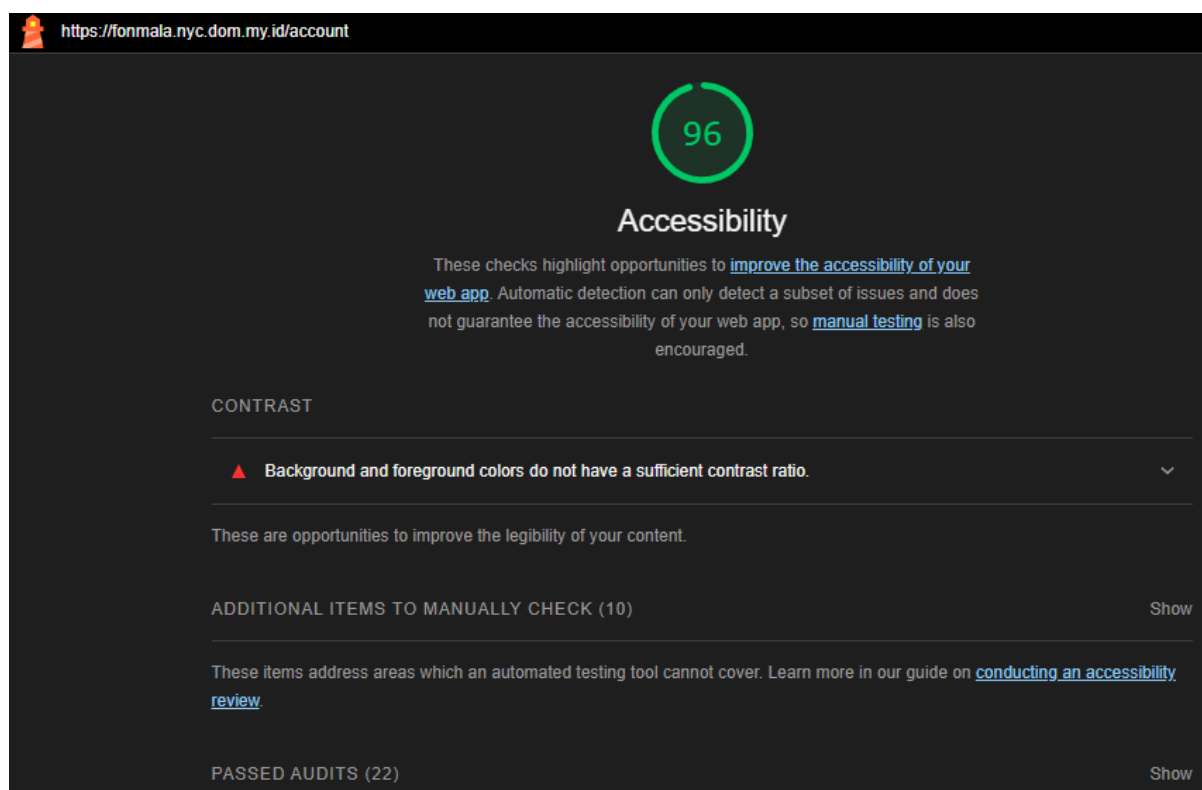


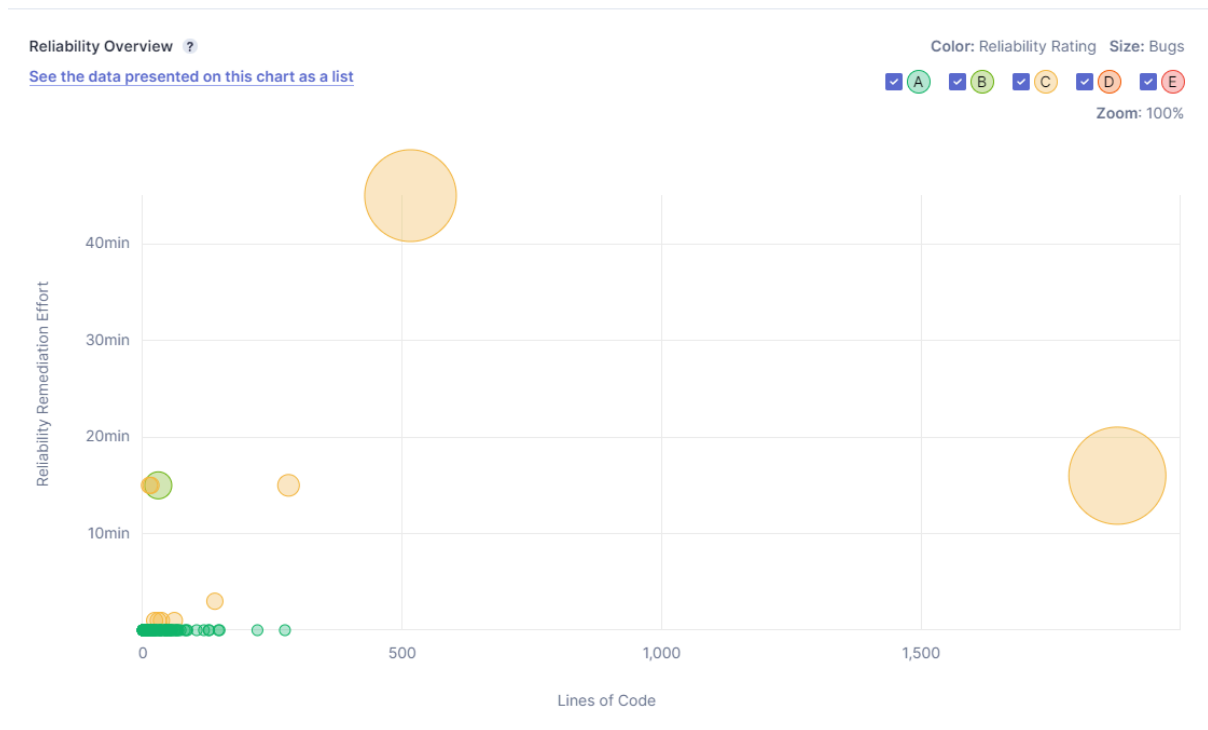
Figura #16 Nota de accesibilidad de la página para ver datos del usuario

Análisis de la prueba de Funcionalidad con 3 hilos

Gestión	Tiempo de ejecución (s)			Resultado	Porcentaje de Éxito
	H1	H2	H3		
Autenticación	44.032	45.169	45.328	Exitosa	100%
Estudiantes	74.650	75.361	75.369	Exitosa	100%
Profesores	24.285	24.306	24.443	Exitosa	100%
Asistentes	37.045	37.125	37.132	Exitosa	100%
Plan de Trabajo	20.213	20.547	20.833	Exitosa	100%
Equipo de Trabajo	15.990	15.992	16.011	Exitosa	100%
Actividades	36.873	37.557	38.169	Exitosa	100%

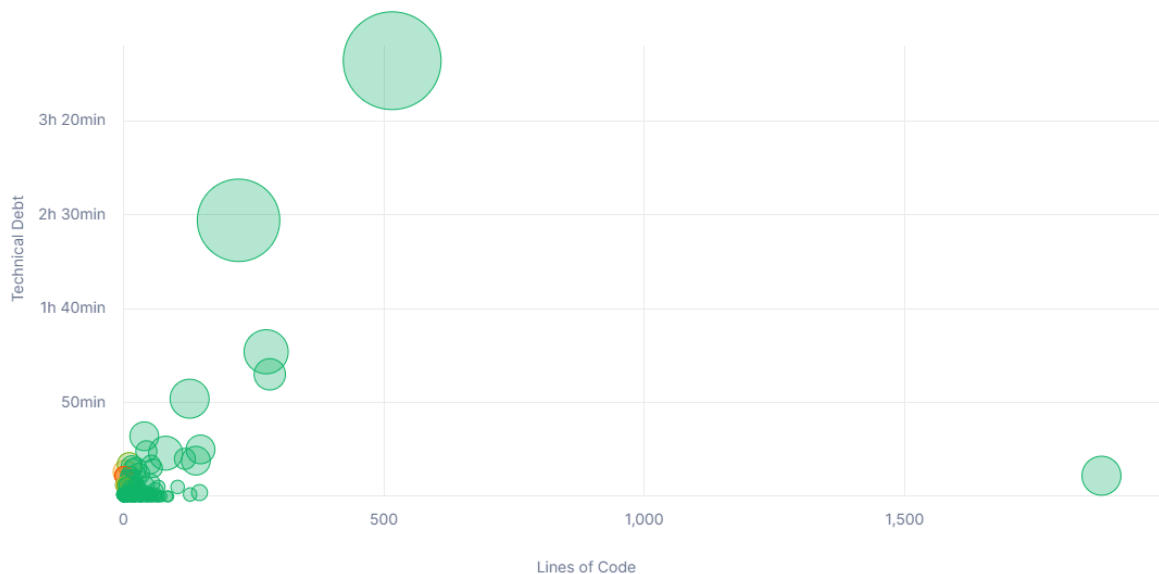
Observaciones: Todas las pruebas se completaron con éxito. No se encontraron errores críticos en las funcionalidades probadas.

Análisis de la deuda técnica



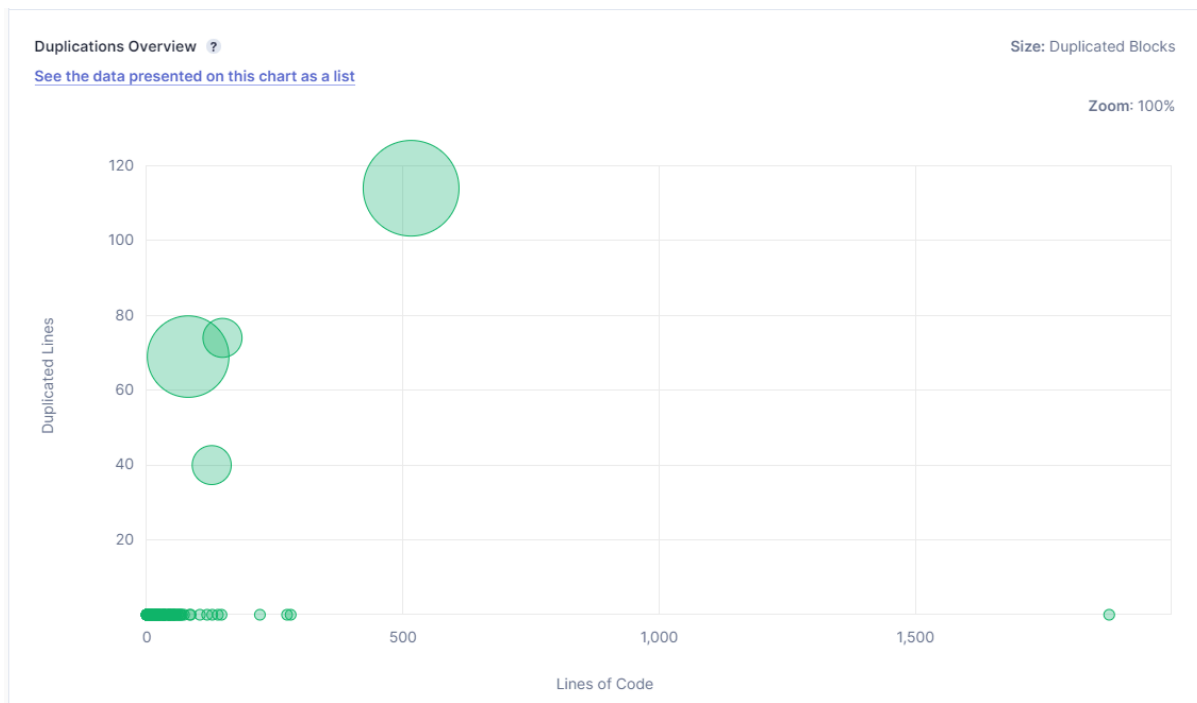
Respecto a la fiabilidad del código analizado, SonarQube nos presenta un gráfico donde en el eje horizontal se muestran las líneas de código, mientras que en el eje vertical se representa el esfuerzo de remediación en términos de tiempo, expresado en minutos. Cada burbuja en el gráfico simboliza una sección de código con un número determinado de bugs, donde el tamaño de la burbuja es proporcional a la cantidad de bugs identificados.

La mayoría de los puntos se encuentran en el extremo izquierdo del gráfico, lo que indica que la mayor parte del código presenta pocos problemas de fiabilidad y requerirá menos de 10 minutos para ser corregido. Esto es una señal positiva, ya que implica que la base de código en su mayoría está bien estructurada y no presenta riesgos importantes de fiabilidad. A pesar de esto, existen algunas burbujas grandes en la zona superior derecha del gráfico, lo que revela la presencia de módulos con mayor cantidad de líneas de código y esfuerzos de remediación más altos (alrededor de 40 minutos). Esto significa que, aunque la mayoría del código tiene buena fiabilidad, hay áreas críticas donde los problemas detectados requieren una intervención significativa, posiblemente debido a una mayor acumulación de errores o bugs en esos módulos.



El gráfico anterior representa el análisis de mantenibilidad del código. El eje horizontal muestra las líneas de código, mientras que el eje vertical representa el tiempo necesario para solucionar estos problemas (deuda técnica), expresado en horas y minutos.

La mayoría de las burbujas están concentradas en la parte inferior izquierda, lo que sugiere que gran parte del código presenta baja deuda técnica y requiere menos de 1 hora de esfuerzo para mejorar su mantenibilidad. Sin embargo, algunas burbujas grandes, que aparecen en módulos con mayor cantidad de líneas de código (hasta 1500), tienen un esfuerzo de remediación considerable (más de 3 horas). Estas burbujas indican áreas críticas donde la complejidad o los code smells se acumulan y deben priorizarse para evitar problemas de mantenimiento futuros. En general, el análisis sugiere que la base de código es mayormente mantenible, con algunas secciones que demandan más esfuerzo debido a su tamaño y acumulación de problemas.



Este gráfico de SonarQube muestra una visión general sobre la duplicación de código en el proyecto. El eje horizontal representa las líneas de código, mientras que el eje vertical indica la cantidad de líneas duplicadas. El tamaño de las burbujas refleja los bloques de código duplicado.

El análisis muestra que la mayoría de los archivos tienen pocas líneas duplicadas. Sin embargo, algunos archivos presentan más de 100 líneas duplicadas, lo cual es una señal preocupante para la mantenibilidad del código. Estas áreas duplicadas pueden aumentar los costos de mantenimiento y la complejidad del proyecto, ya que los cambios deben realizarse en múltiples lugares en lugar de en un solo bloque de código.

classes	12
controllers	267
includes	29
models	231
public	0
src	38
views	112
gulpfile.js	6
Router.php	7

El sistema analizado presenta una complejidad ciclomática total de 702, distribuida principalmente entre los controladores (267) y los modelos (231), que juntos representan más del 70% de la complejidad total del sistema. Los controladores muestran la mayor concentración de complejidad, destacando particularmente `GuiasController.php` con 106 puntos, seguido por `PlansController.php` con 60 puntos. El resto de la complejidad se distribuye entre las vistas (112), `src` (38), `includes` (29), y otros componentes menores.

Esta distribución sugiere un sistema de tamaño mediano con una arquitectura MVC donde la lógica de negocio está mayormente centralizada en los controladores en lugar de los modelos, lo cual podría indicar una oportunidad de mejora en la distribución de responsabilidades. Aunque la complejidad total es considerable, la mayoría de los archivos individuales mantienen niveles manejables por debajo de 50 puntos, con excepción de algunos controladores y modelos específicos que podrían beneficiarse de una refactorización para mejorar su mantenibilidad.

En general, el sistema presenta un nivel moderado de deuda técnica, con áreas específicas que requieren atención prioritaria. Aunque la mayoría del código mantiene buenos estándares, existen puntos críticos que podrían afectar la mantenibilidad a largo plazo. Se presentan fortalezas como:

- La mayoría del código presenta baja deuda técnica (menos de 1 hora de remediación)

- Gran parte de los problemas de fiabilidad requieren menos de 10 minutos para ser corregidos
- La mayoría de los archivos mantienen una complejidad ciclomática manejable (menos de 50 puntos)