



UNIVERSIDAD
CATÓLICA
BOLIVIANA

Informe de Resultados

Campero Morales José Antonio

Campohermoso Berdeja Oscar

Carrasco Céspedes Miguel Alejandro

Martínez Acarapi Fabiola Alejandra

Montero Garrido Diana Aneliz

Zizold Sempertegui Gabriela Zulema Britta

Universidad Católica Boliviana

SIS-312: Gestión de Calidad de Sistemas

Lic. Cecilia Alvarado Monrroy

28 de octubre de 2024

Índice

Resumen de la prueba	1
Evaluación de la prueba	1
Desviaciones de Plan de Prueba	1
Impedimentos de prueba y soluciones alternativas	5
Reporte y Resultados	7
Resultados de las Métricas del Plan de Pruebas	7
Riesgos no mitigados, defectos no corregidos	10
Evaluación de riesgos	10
Riesgos del Producto	10
Riesgos de Proyecto	13
Evaluación de defectos	15
Accesibilidad y usabilidad	16
Resultados Obtenidos en Pruebas de Usabilidad	16
Conclusiones	17
Lecciones Aprendidas y Recomendaciones	18
Anexos	19
Anexo A: Detalle Pruebas Ejecutadas	19
Anexo B - Bugs identificados	70
Anexo C - Bugs Resueltos	94
Anexo D - Reporte Análisis de Accesibilidad	95
Anexo E: Reporte de Defectos de Usabilidad	97

Índice de figuras

1.	Cronograma del Proyecto Final: Diagrama de Gantt	5
2.	Dashboard de Métricas del Plan de Pruebas	9

Índice de tablas

1.	Cronograma Final del Proyecto	4
2.	Tiempo Medio de Reparación de Bugs	8
3.	Plantilla de Reporte de Usabilidad	97

Resumen de la prueba

Durante el proceso de pruebas, se ejecutaron 40 casos de prueba que abarcaron las funcionalidades clave del sistema descritas en las 33 historias de usuario existentes. Del total de casos de prueba, 33 fueron aceptados sin observaciones, cumpliendo con los criterios de aceptación establecidos y asegurando el correcto funcionamiento esperado. Sin embargo, 9 casos fallaron debido a problemas encontrados durante la ejecución, los cuales fueron documentados y reportados para su resolución.

A lo largo de las pruebas, se identificaron 30 defectos en los distintos módulos del sistema, estos se documentaron incluyendo detalles de reproducción, evidencias y sugerencias para su corrección, abarcando tanto aspectos funcionales como de accesibilidad.

Además, se identificaron 7 alertas de accesibilidad, agrupadas según el criterio WCAG afectado, lo cual refleja la evaluación de la accesibilidad del sistema. Todos los reportes mencionados (Test Cases, reporte de errores y alertas de accesibilidad) se encuentran en los anexos.

Evaluación de la prueba

Desviaciones de Plan de Prueba

Las desviaciones en el plan de prueba se originaron principalmente por la coincidencia de los exámenes finales universitarios de los miembros del equipo con las fechas del cronograma del proyecto. Este cruce provocó un periodo de inactividad, lo que afectó la disponibilidad del equipo y causó retrasos en las tareas clave. Esta situación fue temporal y se produjo en la fase inicial e intermedia del proyecto, lo que ocasionó un retraso en varias actividades clave. A continuación, se detallan las desviaciones por tarea:

■ Creación del Repositorio

- **Plan original:** La creación del repositorio estaba prevista para el 2 de diciembre, con una duración estimada de 1.5 horas.

- **Realidad:** La tarea se retrasó un día debido a la falta de disponibilidad de los miembros del equipo, resultando en la ejecución el 3 de diciembre con un total de 2 horas.

■ Documentación base en LaTeX

- **Plan original:** Se planificó para el 2 de diciembre, con una duración de 2.5 horas.
- **Realidad:** Al igual que la tarea anterior, esta se retrasó un día, realizándose el 4 de diciembre, con una duración de 2 horas.

■ Organización del Trabajo

- **Plan original:** Estaba prevista para el 1 de diciembre, con una duración de 2.5 horas.
- **Realidad:** Debido al periodo de inactividad, se pospuso para el 10 de diciembre con una duración de 3 horas, pero no afectó significativamente el flujo de trabajo debido a la flexibilidad del equipo.

■ Pruebas Automatizadas

- **Plan original:** Se pensaba realizar las pruebas desde el 4 hasta el 6 de diciembre, con una duración estimada de 16 horas.
- **Realidad:** Las pruebas se realizaron entre el 11 y el 13 de diciembre, con una duración de 20 horas.

■ Documentación Final

- **Plan original:** La documentación debía completarse entre el 9 y el 10 de diciembre, con una duración de 8 horas.

- **Realidad:** Debido a los retrasos previos, la documentación final, incluyendo sus revisión y ajustes finales, se completó entre el 12 y el 15 de diciembre, con una duración total de 11 horas.

Estas desviaciones fueron causadas por la coincidencia de las responsabilidades académicas de los miembros del equipo con el cronograma del proyecto. Sin embargo, se logró ajustar el plan para minimizar el impacto y garantizar que todas las actividades se completaran dentro de un plazo razonable.

Cronograma Final del Proyecto

El cronograma final obtuvo 38 horas para el periodo de trabajo (3 de diciembre al 15 de diciembre).

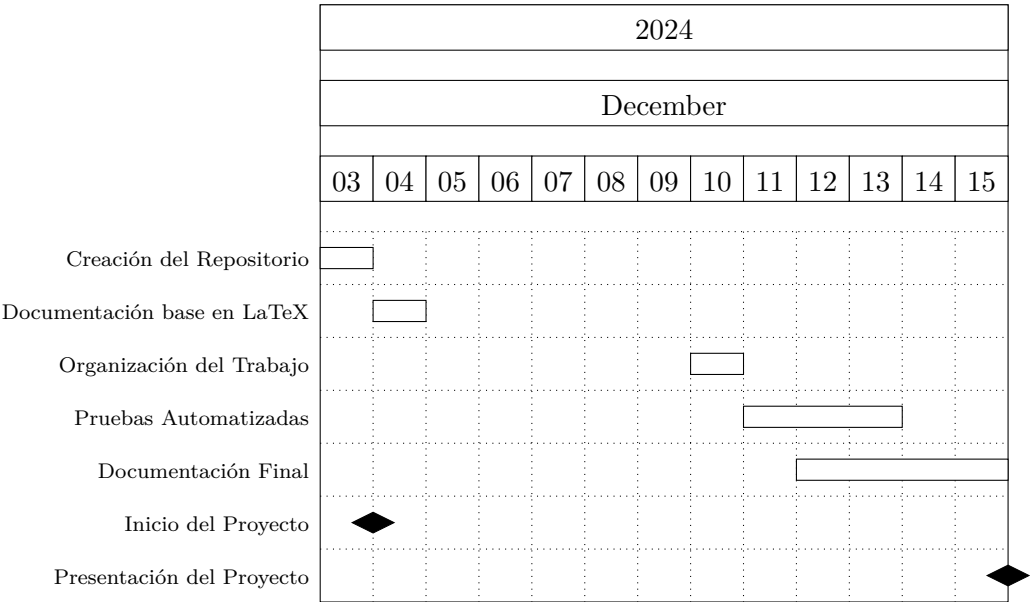
Cuadro 1

Cronograma Final del Proyecto

Actividad	Descripción	Fecha realizada	Duración (d)	Duración (h)
Creación del Repositorio	Configuración inicial para pruebas y documentación.	2024-12-03	1	2
Documentación base en LaTeX	Subida de documentos base para el plan y el informe.	2024-12-04	1	2
Organización del Trabajo	Reunión del equipo para definir tareas.	2024-12-10	1	3
Pruebas Automatizadas	Realización de pruebas (3 Postman, 3 Playwright).	2024-12-11 - 2024-12-13	3	20
Documentación Final	Completar y actualizar los documentos en LaTeX.	2024-12-12 - 2024-12-15	4	11

Diagrama de Gantt Final

Figura 1
Cronograma del Proyecto Final: Diagrama de Gantt



Impedimentos de prueba y soluciones alternativas

A lo largo de las pruebas, se presentaron diversos impedimentos que requirieron soluciones alternativas para garantizar el avance eficiente del proyecto. A continuación, se detallan los principales problemas identificados y las soluciones implementadas:

1. Familiaridad limitada con el sistema

Algunos miembros del equipo que realizaron las pruebas no participaron en el desarrollo del sistema, lo que inicialmente limitó su conocimiento sobre la estructura y el funcionamiento detallado del proyecto. Esta falta de familiaridad hizo necesaria una etapa adicional de recopilación y análisis de las historias de usuario, con el fin de comprender los procesos internos y flujos específicos del sistema.

Solución implementada: Se recurrió a la documentación técnica existente y se organizaron reuniones frecuentes con los desarrolladores y responsables del proyecto para acelerar la comprensión del sistema.

2. Problemas con la configuración de LaTeX

Algunos miembros del equipo encontraron dificultades iniciales relacionadas con la instalación y el aprendizaje del uso de LaTeX para la documentación del informe. Esto ocasionó pequeños retrasos en la preparación de los documentos.

Solución implementada: Se compartieron tutoriales y guías prácticas, además de realizar sesiones de consulta para resolver dudas sobre el uso de LaTeX, lo que permitió al equipo completar la documentación de manera efectiva.

3. Cruce con exámenes académicos

La coincidencia de las actividades del proyecto con los exámenes finales de los miembros del equipo afectó la disponibilidad de los integrantes en momentos clave. Este cruce provocó algunos retrasos en la ejecución de tareas y reuniones.

Solución implementada: Las reuniones se reorganizaron de manera más flexible, priorizando una mejor gestión del tiempo y la organización del equipo para evitar descuidar las actividades del proyecto. Este enfoque permitió cumplir con los objetivos sin comprometer los resultados.

4. Modificaciones en el código para pruebas de interfaz (Playwright)

Durante la implementación de las pruebas automatizadas en Playwright, se detectaron limitaciones en los identificadores del frontend que hacían los localizadores excesivamente largos y complicados. Además, ciertos componentes del frontend mostraban lentitud al ejecutarse en navegadores como WebKit.

Solución implementada: Se realizaron ajustes en el frontend, como la adición de identificadores (`id`) a componentes específicos y alertas para facilitar las pruebas. También se recomendó aumentar el número de trabajadores para mejorar el rendimiento de las pruebas con el comando:

```
npx playwright test tests --workers 6
```

Reporte y Resultados

Resultados de las Métricas del Plan de Pruebas

Los resultados del plan de pruebas reflejan los siguientes indicadores clave:

- **Número total de Test Cases ejecutados:** Se ejecutaron 40 casos de prueba, de los cuales el **82.5 %** (33 casos) fueron exitosos, mientras que 7 resultaron fallidos.
- **Defectos identificados:** Durante las pruebas, se encontraron un total de 30 defectos. De estos, se corrigieron 3, representando un **10 % de bugs resueltos**.
- **Tiempo total de ejecución:** Las pruebas requirieron un total de 15 horas-persona.
- **Cobertura de pruebas:**
 - Historias de Usuario cubiertas: 26
 - Historias de Usuario no cubiertas: 7
- **Automatización de Test Cases:**
 - Usando **Postman**: 4 casos exitosos, 0 fallidos.
 - Usando **Playwright**: 6 casos exitosos, 2 fallidos.

Respecto a los defectos, el análisis del *Tiempo Medio de Reparación (TMR)* muestra que el promedio fue de **33 días y 8 horas**. La siguiente tabla detalla los defectos críticos identificados, sus fechas de resolución y el tiempo requerido para su corrección:

Cuadro 2

Tiempo Medio de Reparación de Bugs

Código	Descripción	Fecha detección	Fecha resolución	TMR (días)
BG-JH-01	Cuando se crea una arista conectada a sí misma, su peso no se mueve junto al elemento asociado	2024-03-15	2024-06-01	78
BG-JH-02	Cuando se intenta solucionar el algoritmo de Johnson con un grafo incorrecto no sucede nada	2024-10-15	2024-10-27	11
BG-JH-03	Cuando se intenta solucionar un grafo con un bucle, el algoritmo entra en un loop infinito	2024-10-15	2024-10-27	11

En términos de accesibilidad, se llevaron a cabo análisis utilizando *Axe Dev Tools* en las vistas de selección de archivos y el editor de grafos Johnson.

- **Antes de las correcciones:** Se detectaron 16 alertas.
- **Después de las correcciones:** Las alertas se redujeron a 7, de las cuales 6 están relacionadas con limitaciones del framework del proyecto, quedando fuera del alcance de esta fase.

Estos resultados están resumidos visualmente en el dashboard presentado a continuación:

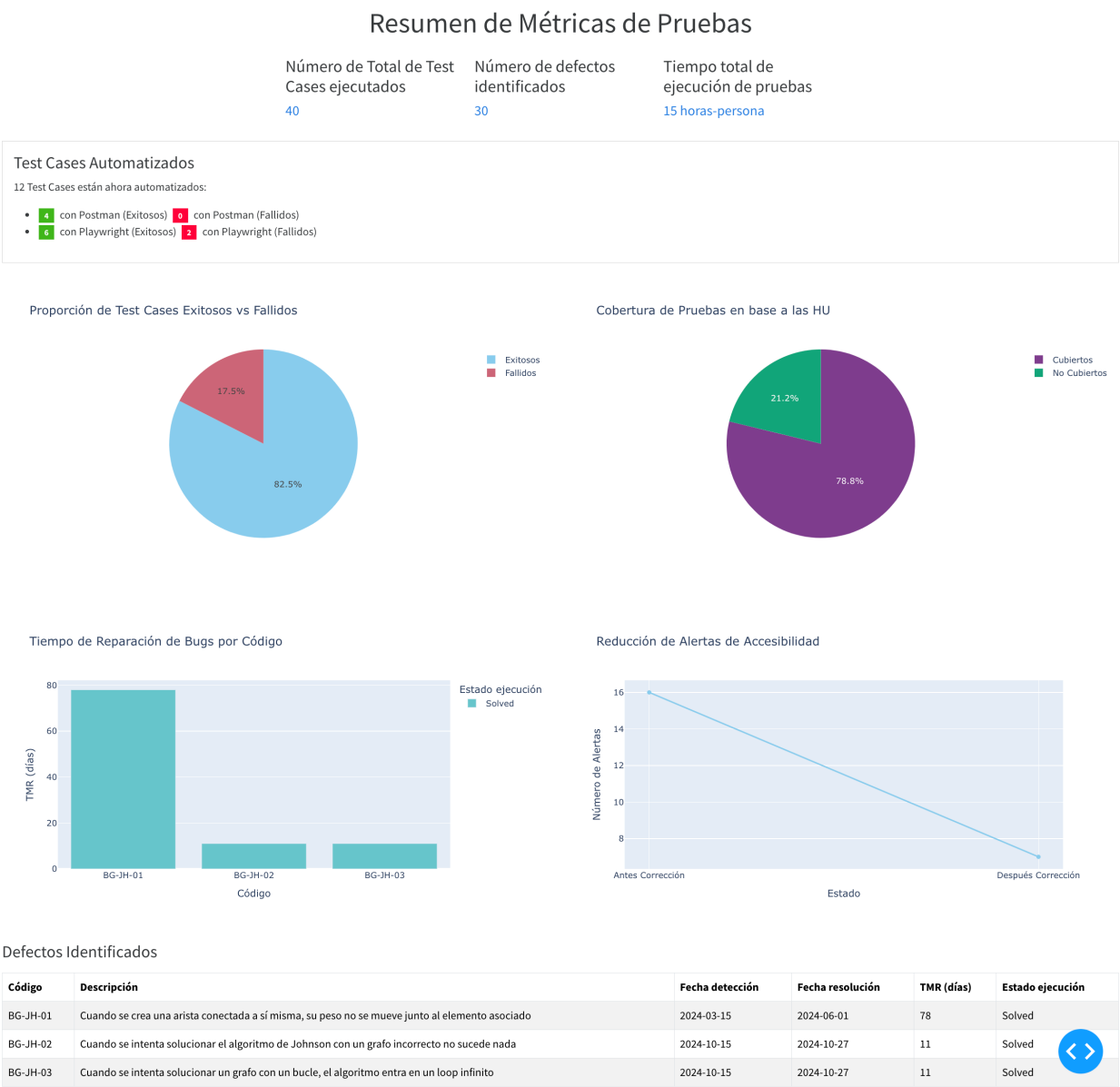


Figura 2

Dashboard de Métricas del Plan de Pruebas

Riesgos no mitigados, defectos no corregidos

Evaluación de riesgos

Durante el proceso de pruebas, algunos de los riesgos previamente identificados impulsaron la creación de pruebas con el objetivo de evaluar la posibilidad de mitigarlos de manera más precisa. Con el reporte de defectos resultante de estas pruebas, se espera implementar acciones correctivas que permitan reducir el impacto y probabilidad de estos riesgos en el sistema.

Después de la evaluación completa del sistema y la ejecución de todas las pruebas planificadas, no se identificaron riesgos adicionales. Los riesgos previamente identificados se abordaron de forma adecuada mediante la implementación de controles y la ejecución de pruebas específicas, lo que permitió mitigar su impacto y probabilidad de ocurrencia.

Riesgos del Producto

■ Resultados inexactos en algoritmos o cálculos complejos

Riesgo: Los algoritmos de cálculo pueden devolver resultados incorrectos debido a errores en la lógica.

Cobertura: Totalmente cubierto

Test Cases relevantes:

- TC-AG-02: Verifica la correcta optimización por pesos máximos.
- TC-AG-03: Verifica la correcta optimización por pesos mínimos.
- TC-DJ-01: Verifica que Dijkstra calcule correctamente el camino mínimo.
- TC-JH-02: Valida el cálculo de la ruta crítica con Johnson.

Riesgo cubierto: Sí, la cobertura es adecuada.

■ Fallas en la comunicación entre backend y frontend

Riesgo: La comunicación entre la interfaz y el servidor puede fallar, generando errores de sincronización de datos.

Cobertura: Totalmente cubierto

Test Cases relevantes:

- TC-PT-01: Valida la correcta respuesta del endpoint de la matriz de adyacencia.
- TC-PT-02: Verifica la respuesta correcta del endpoint de optimización de asignación con maximización.
- TC-PT-03: Valida la correcta respuesta del endpoint del camino crítico con Johnson.
- TC-PT-04: Verifica la respuesta correcta del endpoint de optimización de asignación con minimización.

Riesgo cubierto: Sí, la cobertura es adecuada.

■ Manejo inadecuado de datos en la interfaz del usuario

Riesgo: La interfaz puede no validar correctamente los datos ingresados por el usuario, causando errores.

Cobertura: Totalmente cubierto

Test Cases relevantes:

- TC-PW-02: Verifica el comportamiento al registrar una cuenta con campos vacíos.
- TC-PW-03: Verifica que no se permita registrar cuentas con datos inválidos.
- TC-PW-05: Verifica el inicio de sesión con campos vacíos.
- TC-PW-06: Verifica que no se permita iniciar sesión con datos inválidos.
- TC-NW-05: Valida la interfaz intuitiva y la correcta validación de campos obligatorios.

Riesgo cubierto: Sí, la cobertura es adecuada.

■ **Problemas en el almacenamiento de archivos al guardar un grafo**

Riesgo: La funcionalidad de guardar o cargar grafos no funciona correctamente, y se pueden perder datos.

Cobertura: Totalmente cubierto

Test Cases relevantes:

- TC-GF-03: Verifica que se pueda guardar la estructura del grafo en JSON y restaurar la información.
- TC-JH-12: Verifica la correcta carga y almacenamiento de grafos de Johnson.

Riesgo cubierto: Sí, la cobertura es adecuada.

■ **Rendimiento insuficiente del sistema con grandes volúmenes de datos (Grafos grandes)**

Riesgo: El sistema puede experimentar lentitud o fallos al procesar grandes grafos.

Cobertura: No cubierto

Test Cases relevantes: Ninguno. Este riesgo no está contemplado en los casos de prueba actuales. **Riesgo cubierto:** No, la cobertura es insuficiente.

■ **Cumplimiento insuficiente de estándares de accesibilidad**

Riesgo: La aplicación puede no ser accesible para usuarios con discapacidades.

Cobertura: Cobertura adecuada

Test Cases relevantes: Se realizó una evaluación de accesibilidad con Axe DevTools para cubrir los estándares WCAG 2 AA. **Riesgo cubierto:** Si, la cobertura es suficiente, aunque se identificaron problemas de accesibilidad.

■ **Vulnerabilidades en la seguridad de la gestión de datos y autenticación**

Riesgo: Usuarios no autorizados podrían acceder al sistema o manipular datos sensibles.

Cobertura: Cobertura adecuada

Test Cases relevantes:

- TC-PW-01: Verifica el registro de usuario con datos válidos.
- TC-PW-04: Verifica que los usuarios puedan iniciar sesión con credenciales correctas.
- TC-PW-06: Verifica que no se permita iniciar sesión con datos inválidos.

Riesgo cubierto: Sí, la cobertura es adecuada, aunque puede mejorarse otro tipo de pruebas enfocadas en la seguridad.

Riesgos de Proyecto

- **Demoras en la comunicación y cronograma**

Riesgo: Las demoras en la comunicación entre los equipos de desarrollo y calidad, así como las dificultades en el cumplimiento de los plazos establecidos, pueden retrasar la implementación y las pruebas del sistema de grafos.

Cobertura: Cobertura adecuada

Motivo: La participación de 2 desarrolladores en el equipo de calidad permitió una comunicación más efectiva, reduciendo los retrasos.

Controles:

- Establecimiento de canales de comunicación claros y frecuentes entre los equipos de desarrollo y calidad.
- Uso de herramientas de gestión de proyectos para hacer un seguimiento eficiente de los plazos y las tareas.

Riesgo cubierto: *Sí, la cobertura fue adecuada.*

- **QAs desconocen la lógica interna del sistema**

Riesgo: Los QAs pueden no comprender completamente la lógica de los algoritmos y el sistema de grafos, lo que puede llevar a pruebas incompletas o incorrectas.

Cobertura: Cobertura adecuada

Motivo: Se realizaron reuniones y comunicación directa entre los QAs externos y los QAs que también formaban parte del equipo de desarrollo, logrando una mejor comprensión del sistema.

Controles:

- Documentación clara de la lógica interna de los algoritmos y su funcionamiento dentro del sistema.
- Sesiones de capacitación y colaboración entre desarrolladores y QAs para garantizar la comprensión mutua del sistema.

Riesgo cubierto: *Sí, la cobertura fue adecuada.*

■ Dependencia de QAs hacia developers para el entorno contenedorizado

Riesgo: La dependencia de los QAs hacia los desarrolladores para la configuración y el manejo del entorno contenedorizado puede generar cuellos de botella en las pruebas y la integración continua.

Cobertura: Cobertura parcial

Motivo: Se explicó a los QAs en las reuniones iniciales cómo levantar los entornos, reduciendo la dependencia inicial, aunque aún se requiere cierto apoyo ocasional.

Controles:

- Capacitación a los QAs en la gestión y configuración de entornos contenedorizados, utilizando herramientas como Docker.
- Creación de scripts y documentación detallada para la configuración y despliegue de entornos de prueba.

Riesgo cubierto: *Parcialmente mitigado.*

■ Falta de documentación del sistema

Riesgo: La ausencia de documentación técnica adecuada por parte de los desarrolladores puede dificultar la comprensión, mantenimiento y escalabilidad del sistema de grafos.

Cobertura: Cobertura parcial

Motivo: Se reunieron historias de usuario mediante pruebas exploratorias que se confirmaron con los desarrolladores, aunque se reconoce la necesidad de una documentación más formal.

Controles:

- Revisión de código periódica y mantenimiento de documentación actualizada sobre la arquitectura y algoritmos del sistema.
- Implementación de un sistema de documentación automática, como comentarios detallados en el código y herramientas como Javadoc o Swagger.

Riesgo cubierto: *Parcialmente mitigado.*

Evaluación de defectos

Tras la ejecución de las pruebas, se identificaron un total de 30 defectos en el sistema. Estos defectos han sido documentados y reportados de manera detallada en la sección de anexos de este documento. La atención a estos defectos es fundamental para garantizar el correcto funcionamiento y la confiabilidad del sistema en futuras evaluaciones.

Accesibilidad y usabilidad

Las pruebas de accesibilidad realizadas con Axe DevTools revelaron varios problemas de accesibilidad, aunque en general, la cantidad de defectos encontrados no fue excesiva, lo cual es un indicio de que el sistema ya cuenta con una base relativamente sólida en términos de accesibilidad. Las pruebas que se llevaron a cabo se encuentran detalladas en el reporte de accesibilidad adjunto en anexos.

Resultados Obtenidos en Pruebas de Usabilidad

Las pruebas de usabilidad realizadas permitieron evaluar varios aspectos clave del sistema, como la visibilidad del estado, el control y la libertad del usuario, la consistencia y estándares, entre otros. En general, se detectaron áreas de mejora en términos de consistencia visual, retroalimentación al usuario, y navegación, los cuales se detallan en los resultados adjuntos. Cabe destacar que el sistema cumple con algunos de los principios fundamentales, como la flexibilidad y la visibilidad en varias de sus opciones, lo cual es positivo para la experiencia de usuario.

Para una descripción completa de los resultados obtenidos en cada ítem evaluado, por favor consulte los detalles (en el Anexo A)

Conclusiones

- **Evaluación general del sistema:** Los resultados obtenidos en las pruebas indican que el sistema cumple en gran medida con las funcionalidades especificadas, especialmente en los módulos clave como los algoritmos Compet y Dijkstra. Sin embargo, también se identificaron debilidades en la resistencia a casos atípicos, accesibilidad y retroalimentación visual. Estas áreas necesitan mejoras para garantizar un funcionamiento más robusto y alineado con los estándares de calidad esperados.
- **Validación de datos y manejo de errores:** Durante las pruebas se detectaron defectos en la validación de campos obligatorios y numéricos, así como en la notificación de errores en entradas inconsistentes. Estos problemas afectan la precisión de los cálculos y la experiencia del usuario. La implementación de mecanismos de validación más rigurosos y notificaciones específicas es esencial para corregir estas deficiencias.
- **Accesibilidad y usabilidad:** Las pruebas de accesibilidad revelaron incumplimientos con los criterios WCAG 2.2, como contraste inadecuado de colores y falta de etiquetado descriptivo para lectores de pantalla. Aunque se realizaron correcciones parciales, se recomienda implementar verificaciones recurrentes de accesibilidad para mejorar la experiencia de todos los usuarios, incluyendo aquellos con discapacidades.
- **Resultados de pruebas clave:** De los casos de prueba ejecutados, varios fueron aceptados sin observaciones, confirmando que los módulos cumplen con los criterios de aceptación. Sin embargo, algunos defectos de baja prioridad y otros más significativos, como problemas en la visualización de listas extensas y superposiciones en el módulo de Sorts, fueron documentados para futuras iteraciones.
- **Gestión de riesgos y defectos:** Se identificaron seis defectos importantes que, aunque no se corrigieron en su totalidad, han sido documentados detalladamente. La

falta de ajustes automáticos en situaciones de desbalance entre oferta y demanda también representa un riesgo crítico que debe priorizarse en las próximas fases de desarrollo.

- **Conclusión final:** Si bien el sistema demuestra estabilidad en sus módulos principales y cumple con los requisitos funcionales, las áreas de accesibilidad, usabilidad y documentación requieren atención significativa. Las mejoras sugeridas no solo aumentarán la calidad general del producto, sino que también garantizarán una experiencia de usuario inclusiva y satisfactoria.

Lecciones Aprendidas y Recomendaciones

- **Lecciones aprendidas:**

- La necesidad de pruebas tempranas y recurrentes en accesibilidad y usabilidad resultó ser una lección clave para asegurar la inclusividad desde las etapas iniciales del desarrollo.
- La falta de estimaciones precisas en el tiempo necesario para la ejecución del plan de pruebas resaltó la importancia de una mejor gestión de tiempos en proyectos futuros.
- La documentación limitada del proceso de desarrollo dificultó la replicabilidad de casos y la resolución eficiente de defectos.

- **Recomendaciones para mejoras:**

- Implementar una suite de pruebas automatizadas para cubrir los endpoints y simplificar el testing continuo de la API.
- Mejorar el proceso de validación de datos, asegurando que todos los campos requeridos tengan validaciones adecuadas y mensajes de error comprensibles.
- Optimizar la accesibilidad mediante contrastes adecuados de colores, etiquetas descriptivas para lectores de pantalla y simplificación de la navegación.

- Realizar pruebas de carga y escalabilidad en módulos como Sorts para asegurar el correcto funcionamiento con grandes volúmenes de datos.
- Priorizar la corrección de defectos identificados en las pruebas, especialmente aquellos relacionados con accesibilidad y riesgos funcionales.
- Asegurar que futuras iteraciones incluyan documentación más detallada tanto del proceso de desarrollo como del plan de pruebas, facilitando su comprensión y mantenimiento.

Anexo A - Detalle Pruebas Ejecutadas

Aa Codigo	Descripción	Severidad	Precondiciones	Resultado esperado	Evidencias	Estado ejecución
<u>TC-ST-03</u>	Verificar que el sistema muestre una animación paso a paso del proceso de ordenamiento.	Medium	- El usuario debe haber seleccionado un algoritmo de ordenamiento. - La opción de visualización debe estar activada.		<u>Loom Evidencias</u>	Passed
<u>03-01</u>	El usuario accede la módulo de sorts			El módulo de Sorts se carga correctamente, mostrando todas las opciones disponibles para generar y ordenar números.		
<u>03-02</u>	El sistema muestra la animación del ordenamiento paso a paso.			El sistema presenta la animación del algoritmo seleccionado de manera clara y progresiva, mostrando cómo los elementos de la lista se comparan, intercambian (si corresponde), y avanzan hacia su		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				posición ordenada en tiempo real.		
<u>TC-ST-02</u>	Verificar que el sistema permita generar una lista de números aleatorios para utilizar en los algoritmos de ordenamiento.	High	- El usuario debe tener acceso al módulo de Sorts.		<u>TCST01.png</u> , <u>TCST02.png</u> , <u>TCST03.png</u>	Failed
<u>02-01</u>	El usuario accede al módulo de Sorts.			El módulo de Sorts se carga correctamente, mostrando todas las opciones disponibles para generar y ordenar números.		
<u>02-02</u>	El usuario selecciona la opción para generar una lista de números aleatorios.			La opción para generar números aleatorios se activa sin errores y permite al usuario proceder al siguiente paso.		
<u>02-03</u>	El usuario define la cantidad de números que desea generar (por ejemplo, 10, 50, 100).			El sistema registra la cantidad definida por el usuario (por ejemplo, 10, 50 o 100) y valida que sea un número válido y dentro del rango permitido.		
<u>02-04</u>	El usuario confirma la generación de números aleatorios.			La generación de números aleatorios se confirma sin		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				problemas, y el sistema comienza el proceso de creación.		
<u>02-05</u>	El sistema muestra la lista de números generados en la pantalla.			La lista de números generados se muestra correctamente en la pantalla, reflejando la cantidad solicitada y con números aleatorios válidos.		
<u>02-06</u>	El usuario selecciona uno de los algoritmos de ordenamiento disponibles (por ejemplo, Selection Sort) y ejecuta el ordenamiento en la lista generada.1.			El algoritmo de ordenamiento seleccionado se ejecuta correctamente, mostrando paso a paso cómo los números en la lista son ordenados según el método elegido.		
<u>TC-ST-01</u>	Verificar que el sistema ordena una lista de números proporcionada por el usuario usando el método seleccionado.	Medium	<ul style="list-style-type: none">- El usuario debe tener acceso al módulo de Sorts.- El usuario debe haber iniciado sesión en la aplicación.		<u>TCST01.png</u>	Passed

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	⬇️ Estado ejecución
<u>01-01</u>	El usuario ingresa a la vista de Sorts.			La vista de Sorts se carga correctamente, mostrando los controles para ingresar números, seleccionar algoritmos y ejecutar el ordenamiento.		
<u>01-02</u>	El usuario ingresa una lista de números manualmente o carga un archivo con números.			Los números ingresados manualmente o cargados desde un archivo se muestran en pantalla de manera clara y sin errores. Si el archivo contiene datos inválidos, se muestra un mensaje de error específico.		
<u>01-03</u>	El usuario selecciona uno de los algoritmos de ordenamiento disponibles.			El algoritmo de ordenamiento seleccionado se resalta o indica visualmente, confirmando la elección del usuario.		
<u>01-04</u>	El usuario inicia el proceso de ordenamiento.			El proceso de ordenamiento inicia correctamente,		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				mostrando cada paso del algoritmo seleccionado en la visualización interactiva hasta completarse.		
<u>TC-</u> <u>PW-08</u>	(Automatizado en Playwright) Verificar que se generen números aleatorios, se realice un ordenamiento por selección (Selection Sort) y se valide que la salida esté ordenada correctamente.	High	- Todos los servicios necesarios (Docker, backend, frontend) deben estar operativos. - El usuario debe estar registrado.		<u>PWSortsTestsResults.png</u>	Passed
<u>08-01</u>	Ingresar a la ruta <code>/editor_sorts</code>			Se muestra la vista vacía del editor de ordenamientos.		
<u>08-02</u>	Seleccionar el botón "Generar números aleatorios"			Aparece un modal para configurar la cantidad de números.		
<u>08-03</u>	Ingresar la cantidad de números aleatorios a generar (ejemplo: 40)			Los números aleatorios aparecen en pantalla.		
<u>08-04</u>	Configurar el delay de animación a "0"			La configuración se actualiza exitosamente.		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>08-05</u>	Seleccionar el botón "Ordenar por Selection Sort"			Inicia el proceso de ordenamiento.		
<u>TC- PW-07</u>	(Automatizado en Playwright) Verificar que se genere un arbol binario válido a partir de sus recorridos en pos-orden e in-orden	High	- Todos los servicios necesarios (Docker, backend, frontend) deben estar operativos. - El usuario debe estar registrado		<u>TCPW03.png</u>	Passed
<u>07-01</u>	Ingresar a la ruta <code>/editor-binary-trees</code>			Se muestra la vista vacía del editor de árboles		
<u>07-02</u>	Seleccionar el botón <code>Ingresar datos</code>			Aparece un modal para ingresar dos listas de datos, el recorrido en in-orden y en pos-orden		
<u>07-03</u>	Generar un árbol binario y extraer sus tres recorridos, ingresar los recorridos in-orden y pos-orden a los campos correspondientes y presionar OK			Se muestra en el canvas un diagrama con el árbol binario estructurado según las rutas ingresadas		
<u>07-04</u>	Presionar el botón con ícono <code>123</code> para visualizar las rutas y validar que los recorridos sean los correctos			Se muestra el modal con los 3 recorridos y estos coinciden con los que se		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				generaron el paso anterior.		
<u>TC-</u> <u>PW-06</u>	(Automatizado en Playwright) Verificar que el sistema no permita iniciar sesión con datos inválidos.	Medium	Todos los servicios necesarios deben estar activos.		<u>PWTestsResults.png</u>	Passed
<u>06-01</u>	Ingresar a la vista de inicio de sesión.		Se muestra la vista vacía del formulario de inicio.			
<u>06-02</u>	Ingresar datos inválidos (correo con formato incorrecto, contraseña incorrecta).		El sistema muestra mensajes de error adecuados.			
<u>06-03</u>	Presionar el botón "Iniciar sesión".		El sistema no permite el inicio y notifica los errores.			
<u>TC-</u> <u>PW-05</u>	(Automatizado en Playwright) Verificar el comportamiento del sistema al intentar iniciar sesión con campos vacíos.	Medium	Todos los servicios necesarios deben estar activos.		<u>PWTestsResults.png</u>	Passed
<u>05-01</u>	Ingresar a la vista de inicio de sesión.		Se muestra la vista vacía del formulario de inicio.			
<u>05-02</u>	Dejar uno o ambos campos (correo, contraseña) vacíos.		El sistema muestra			

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	⌂ Resultado esperado	📎 Evidencias	⌵ Estado ejecución
			mensajes de error específicos.			
<u>05-03</u>	Presionar el botón "Iniciar sesión".		El sistema no permite el inicio y se mantiene en la misma vista.			
<u>TC-PW-04</u>	(Automatizado en Playwright) Verificar que un usuario pueda iniciar sesión ingresando credenciales correctas.	High	- Todos los servicios necesarios deben estar operativos. - El usuario debe tener una cuenta registrada.		<u>PWTestsResults.png</u>	Passed
<u>04-01</u>	Ingresar a la vista de inicio de sesión.		Se muestra la vista vacía del formulario de inicio.			
<u>04-02</u>	Completar los campos (correo y contraseña) con credenciales correctas.		Los campos se completan sin errores.			
<u>04-03</u>	Presionar el botón "Iniciar sesión".		El sistema redirige correctamente a la vista principal.			

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>TC-</u> <u>PW-03</u>	(Automatizado en Playwright) Verificar que el sistema no permita registrar una cuenta con datos inválidos.	Medium	Todos los servicios necesarios deben estar activos.		<u>PWTestsResults.png</u>	Failed
<u>03-01</u>	Ingresar a la vista de registro.			Se muestra la vista vacía del formulario de registro.		
<u>03-02</u>	Completar campos con datos inválidos (contraseña con menos de 6 caracteres).			El sistema muestra mensajes de error específicos.		
<u>03-03</u>	Presionar el botón "Registrar".			La cuenta no se crea y se notifican los errores.		
<u>TC-</u> <u>PW-02</u>	(Automatizado en Playwright) Verificar el comportamiento del sistema al registrar una cuenta con campos vacíos.	Medium	Todos los servicios necesarios deben estar activos.		<u>PWTestsResults.png</u>	Failed
<u>02-01</u>	Ingresar a la vista de registro.		Se muestra la vista vacía del formulario de registro.			
<u>02-02</u>	Dejar uno o varios campos vacíos (por ejemplo, campo correo).		El sistema muestra mensajes de error para cada campo vacío.			
<u>02-03</u>	Presionar el botón "Registrar".		La cuenta no se crea y se			

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	📌 Estado ejecución
			mantiene en la misma vista.			
<u>TC-PW-01</u>	(Automatizado en Playwright) Verificar que un usuario pueda registrarse correctamente ingresando todos los datos válidos.	High	- Todos los servicios necesarios (Docker, backend, frontend) deben estar operativos. - El usuario no debe existir en el sistema.		<u>PWTestsResults.png</u>	Passed
<u>01-01</u>	Ingresar a la vista de registro.			Se muestra la vista vacía del formulario de registro.		
<u>01-02</u>	Completar todos los campos (nombre, correo, contraseña) con datos válidos.			Los campos se completan sin errores.		
<u>01-03</u>	Presionar el botón "Registrar".			La cuenta se crea exitosamente y se redirige al login.		
<u>TC-PT-03</u>	(Automatizado en Postman) Verificar que el endpoint POST http://localhost:8081/algorithms/johnsonCriticalPath procese correctamente el grafo obtenido y devuelva la ruta crítica del algoritmo jhonson	High	- Backend ejecutado en un contenedor. -Frontend del proyecto ejecutado en local. -. Acceso al proyecto a través del		<u>TC-PT-03.png</u>	Passed

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⌵ Estado ejecución
			navegador - Tener una cuenta registrada en la plataforma. - Iniciar sesión con la cuenta registrada.			
<u>03-02</u>	Enviar la solicitud con el cuerpo que contiene el grafo generado previamente.			La respuesta debe devolver un código de estado 200.		
<u>03-01</u>	Crear un grafo aleatorio en el script de pre-ejecución utilizando datos de prueba almacenados en un arreglo de JSONs.			Un grafo aleatorio seleccionado y asignado a la variable de entorno <code>selectedJson</code>		
<u>03-03</u>	Comprobar el tiempo de respuesta de la solicitud.			El tiempo de respuesta debe ser menor a 200ms.		
<u>03-04</u>	Comprobar que el cuerpo de la respuesta sea de tipo JSON.			El cuerpo debe estar en formato JSON válido.		
<u>03-05</u>	Asegurarse de que la respuesta contenga los atributos activities, criticalPathNodes, criticalPathEdges, y criticalPathDuration.			El cuerpo debe contener todos los atributos esperados.		
<u>03-06</u>	Comprobar que cada actividad en activities incluya los atributos requeridos: "edgeId", "duration", "earlyStart", "earlyFinish", "lateStart", "lateFinish", "slack", "critical", "predecessorIds", "successorIds", "originNodeName", "targetNodeName".			odas las actividades deben incluir los atributos requeridos.		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>03-07</u>	Validar que el atributo criticalPathDuration sea un número positivo.			criticalPathDuration debe ser un número mayor a 0		
<u>03-08</u>	Verificar que no haya actividades duplicadas en la lista activities, usando el atributo edgeld.			La lista de actividades no debe contener elementos duplicados.		
<u>TC-PT-02</u>	(Automatizado en Postman) Verificar que el endpoint POST http://localhost:8081//algorithms/assignationOptimization?maximize=true , procese el cálculo de maximización por asignación de un grafo, devolviendo la solución con el formato esperado y los códigos de estado correctos.	High	- Backend corriendo y docker encendido - Usuario creado - Token de autenticación válido. - Se debe disponer de un grafo aleatorio ingresado previamente		PostmanCorrect02.png	Passed
<u>02-01</u>	Generar un grafo aleatorio en el script de pre-ejecución			Un grafo aleatorio seleccionado y asignado a la variable de entorno <code>selectedJson</code>		
<u>02-02</u>	Enviar la solicitud al endpoint POST http://localhost:8081//algorithms/assignationOptimization?maximize=true con el cuerpo que contiene el grafo generado			La respuesta debe devolver un código de estado 200.		
<u>02-03</u>	Verificar que tiempo de respuesta.			El tiempo de respuesta debe ser		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	📌 Estado ejecución
				menor a 200ms.		
<u>02-04</u>	Verificar el formato de la respuesta.			El cuerpo debe estar en formato JSON válido.		
<u>02-05</u>	Verificar formato del cuerpo de la respuesta.			El cuerpo debe contener todos los atributos esperados.		
<u>02-06</u>	Verificar que el atributo "cost" sea un número positivo			El atributo "cost" es un número positivo		
<u>02-07</u>	Verificar que el atributo de "optimization" es "max"			El atributo "optimization" es "max"		
<u>TC-PT-04</u>	(Automatizado en Postman) Verificar que el endpoint POST http://localhost:8081//algorithms/assignationOptimization?maximize=false , procese el cálculo de minimización por asignación de un grafo, devolviendo la solución con el formato esperado y los códigos de estado correctos.	High	- Backend corriendo y docker encendido - Usuario creado - Token de autenticación válido. - Se debe disponer de un grafo aleatorio ingresado previamente		PostmanCorrect02.png	Passed
<u>04-01</u>	Generar un grafo aleatorio en el script de pre-ejecución			Un grafo aleatorio seleccionado y asignado a la variable de entorno <code>selectedJson</code>		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⏏ Estado ejecución
<u>04-02</u>	Enviar la solicitud al endpoint POST http://localhost:8081/algorithms/assignationOptimization?maximize=false con el cuerpo que contiene el grafo generado			La respuesta debe devolver un código de estado 200.		
<u>04-03</u>	Verificar que tiempo de respuesta.			El tiempo de respuesta debe ser menor a 200ms.		
<u>04-04</u>	Verificar el formato de la respuesta.			El cuerpo debe estar en formato JSON válido.		
<u>04-05</u>	Verificar formato del cuerpo de la respuesta.			El cuerpo debe contener todos los atributos esperados.		
<u>04-06</u>	Verificar que el atributo "cost" sea un número positivo			El atributo "cost" es un número positivo		
<u>04-07</u>	Verificar que el atributo de "optimization" es "min"			El atributo "optimization" es "min"		
<u>TC-PT-01</u>	(Automatizado en Postman) Verificar que el endpoint POST http://localhost:8081/graph/adjMatrix procese correctamente la matriz de adyacencia de un grafo aleatorio, devolviendo la respuesta con el formato esperado y los códigos de estado correctos.	High	- Backend corriendo - Token de autenticación válido. - Se debe disponer de un grafo aleatorio generado previamente - La estructura del grafo debe		PostmanCorrect01.png	Passed

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
			tener desde 0 nodos.			
<u>01-01</u>	Generar un grafo aleatorio en el script de pre-ejecución con nodos.			El grafo debe contener un campo de nodos que puede tener 0 nodos.		
<u>01-02</u>	Enviar la petición POST al endpoint http://localhost:8081/graph/adjMatrix con el cuerpo de la solicitud			La respuesta debe devolver un código de estado 200		
<u>01-03</u>	Validar que tiempo de respuesta.			El tiempo de respuesta no debe superar los 200ms.		
<u>01-04</u>	Verificar el formato de la respuesta.			El cuerpo de la respuesta debe ser de tipo JSON		
<u>01-05</u>	Validar formato del cuerpo de la respuesta.			El cuerpo de la respuesta debe contener los siguientes atributos: 'values', 'n', 'verticesIds', 'rowSum', 'colSum', 'mtxSum'		
<u>01-06</u>	Verificar que el atributo 'n' sea un número mayor o igual a 0.			El atributo 'n' debe ser un número y debe ser mayor o igual a 0		
<u>01-07</u>	Validar que el valor de 'n' coincida con la cantidad de elementos en ciertos parámetros.			l atributo 'n' debe coincidir con la cantidad de elementos de		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⌵ Estado ejecución
				'verticesIds' y 'verticesNames'.		
<u>TC-NW-06</u>	Verificar que el editor permita a los usuarios definir nodos y conexiones con pesos, y que los grafos se visualicen de forma intuitiva en la interfaz.	High	- Configuración inicial del editor sin grafos. - Usuarios con capacidad de definir nodos y conexiones.		<u>TC-NW-06.png</u>	Failed
<u>06-01</u>	Definir un nodo en el editor.			El nodo debe aparecer inmediatamente en la visualización gráfica del editor.		
<u>06-02</u>	Ingresar datos válidos en todos los campos y enviar el formulario.			La conexión debe visualizarse en el grafo en tiempo real, mostrando el peso asignado.		
<u>06-03</u>	Intentar crear una conexión con datos inconsistentes (por ejemplo, un peso no numérico).			El sistema debe notificar al usuario sobre la inconsistencia en la entrada de datos, previniendo errores en la creación del grafo.		Failed
<u>06-04</u>	Modificar la conexión o el nodo existente.			La visualización del grafo debe actualizarse de inmediato para		

Aa Codigo	Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇️ Estado ejecución
				reflejar los cambios realizados.		
<u>TC-NW-05</u>	Asegurarse de que el formulario de entrada de datos sea intuitivo y que los campos obligatorios estén correctamente validados.	High	<ul style="list-style-type: none"> - El editor debe estar configurado correctamente. - Los nodos de origen y destino deben poder ser creados y conectados sin errores. - Datos válidos e inválidos en el formulario de entrada, incluyendo campos vacíos y datos no numéricos en los campos de oferta y demanda. 		<u>TC-NW-05.png</u>	Failed
<u>05-01</u>	Intentar enviar el formulario con algunos campos vacíos.			El sistema debe mostrar mensajes de error específicos indicando que los campos obligatorios están vacíos.		Failed
<u>05-02</u>	Ingresar datos no numéricos en los campos de oferta y demanda y enviar el formulario.			El sistema debe mostrar mensajes de error indicando que		Failed

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇️ Estado ejecución
				los valores en oferta y demanda deben ser numéricos.		
<u>05-03</u>	Ingresar datos válidos en todos los campos y enviar el formulario, sin demanda y oferta.			El sistema debe lanzar un mensaje de error porque los datos de demanda y oferta deberían ser obligatorios.		Failed
<u>TC-NW-04</u>	Verificar que el sistema siempre genere una solución factible para diferentes configuraciones de oferta y demanda.	High	<ul style="list-style-type: none">- El editor debe estar configurado correctamente.- Los nodos de origen y destino deben poder ser creados y conectados sin errores.- Los pesos asignados deben ser valores numéricos válidos y compatibles con El sistema.- Varias configuraciones de oferta y demanda, incluyendo		<u>TC-NW-03-2.png, Captura de pantalla 2024-10-28 a la(s) 23.01.24.png</u>	Failed

Aa Codigo	Descripción	Severidad	Precondiciones	Resultado esperado	Evidencias	Estado ejecución
			configuraciones balanceadas y no balanceadas			
<u>04-01</u>	Ingresar diferentes valores de oferta y demanda (balanceados y no balanceados).			El sistema debe aceptar los valores de oferta y demanda sin errores, independientemente de si están balanceados.		
<u>04-02</u>	Ejecutar el cálculo del problema de transporte utilizando el módulo "Northwest".			El sistema debe generar una solución válida y consistente, sin importar si la configuración de oferta y demanda está balanceada.		Failed
<u>TC-NW-03</u>	Verificar que las opciones de maximizar y minimizar generen soluciones distintas.	High	<ul style="list-style-type: none">- El editor debe estar configurado correctamente, permitiendo crear y conectar nodos.- Pesos numéricos válidos y matriz de costos conocida.		<u>TC-NW-03.png</u>	Passed

Aa Codigo	Descripción	Severidad	Precondiciones	Resultado esperado	Evidencias	Estado ejecución
<u>03-01</u>	Configurar la matriz de costos.			El sistema acepta la matriz de costos sin errores y la muestra correctamente.		Passed
<u>03-02</u>	Seleccionar maximización y calcular.			El sistema calcula y muestra la solución sin errores.		Passed
<u>03-03</u>	Verificar la solución en maximización.			Solución visible y accesible en la interfaz.		Passed
<u>03-04</u>	Seleccionar minimización y calcular.			El sistema calcula y muestra la solución sin errores.		Passed
<u>03-05</u>	Comparar ambas soluciones.			Las soluciones deben diferir según el criterio de optimización.		Passed
<u>TC-NW-02</u>	Verificar el comportamiento del sistema cuando la suma de la oferta y la demanda no es igual.	Medium	<ul style="list-style-type: none">- El editor debe estar configurado correctamente.- Los nodos de origen y destino deben poder ser creados y conectados sin errores.- Los pesos asignados deben ser valores		<u>TC-NW-02.png</u>	Failed

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
			numéricos válidos y compatibles con El sistema. - La oferta total debe ser mayor que la demanda.			
<u>02-01</u>	Ingresar 100 unidades de oferta para los nodos de origen			El sistema debe aceptar la cantidad de oferta sin mostrar errores.		Failed
<u>02-02</u>	Ingresar 80 unidades de demanda para los nodos de destino.			El sistema debe aceptar la cantidad de demanda sin mostrar errores.		Failed
<u>02-03</u>	Ejecutar el cálculo utilizando el método de la esquina noroeste.			El sistema debe detectar que la oferta y la demanda no están balanceadas y ajustar automáticamente la solución (por ejemplo, añadiendo un nodo ficticio) para resolver el problema de transporte sin errores.		Failed
<u>TC- NW-01</u>	Comprobar que, al crear los nodos y las conexiones, se genera automáticamente la matriz con los valores	High	- El editor debe estar		<u>TC-NW-01.png</u>	Passed

Aa Codigo	Descripción	Severidad	Precondiciones	Resultado esperado	Evidencias	Estado ejecución
	correctos de los pesos.		configurado correctamente. - Los nodos de origen y destino deben poder ser creados y conectados sin errores. - Los pesos asignados deben ser valores numéricos válidos y compatibles con el sistema.			
<u>01-01</u>	Crear 3 nodos de origen y 3 nodos de destino en el editor.			Los 6 nodos son creados correctamente sin errores.		Passed
<u>01-02</u>	Asignar conexiones entre ellos con pesos específicos para cada enlace.			Las conexiones se establecen correctamente y los pesos asignados son visibles en la interfaz.		Passed
<u>01-03</u>	Abrir el formulario "Northwest" para ver la matriz generada.			La matriz generada muestra los valores de los pesos correctamente, coincidiendo con los valores de las conexiones creadas.		Passed


Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇️ Estado ejecución
<u>TC-KS-02</u>	Verificar que el usuario pueda aplicar el algoritmo de Kruskal en el editor de grafos, seleccionando la opción de maximizar para obtener el árbol de expansión máxima y visualizar tanto la ruta óptima resaltada como el valor de la suma total de la ruta.	High	<ul style="list-style-type: none"> - Backend ejecutado en un contenedor. -Frontend del proyecto ejecutado en local. - Acceso al proyecto a través del navegador - Tener una cuenta registrada en la plataforma. - Iniciar sesión con la cuenta registrada. 		<u>TC-KS-02.png</u>	Passed
<u>02-01</u>	Accede a la aplicación en http://localhost:5173 o en el dominio desplegado.		La aplicación debe abrirse en la página de inicio (home).			
<u>02-02</u>	Dirígete al apartado de "Grafos" desde el navbar o accediendo a http://localhost:5173/graphs_home.		Se debe mostrar una lista de todas las opciones de grafos disponibles, junto con una breve descripción de cada una.			

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>02-03</u>	Selecciona la opción "Grafos".		La aplicación debe redirigir al editor de grafos, mostrando la opción de abrir el editor y una breve descripción del módulo de grafos.			
<u>02-04</u>	Selecciona la opción "Ir a Grafos".		El editor de grafos debe mostrarse en pantalla, con el menú de edición de grafos habilitado para el usuario.			
<u>02-05</u>	En el menú del editor, selecciona la opción "Kruskal".		Debe aparecer una descripción del algoritmo de Kruskal con un botón de "Ir a Kruskal".			
<u>02-06</u>	Presiona "Ir a Kruskal" para acceder a la funcionalidad del algoritmo de Kruskal.		El editor debe cargar con opciones para aplicar el algoritmo de Kruskal.			

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	⌂ Resultado esperado	📎 Evidencias	⌄ Estado ejecución
<u>02-07</u>	Crea un grafo básico en el editor, siguiendo los pasos del test case TC-JAC-M03-001 como referencia.		El grafo debe visualizarse correctamente en el editor con todos los nodos y aristas creados.			
<u>02-08</u>	En el menú de Kruskal, selecciona la opción "Maximizar" en el check de opciones.		El sistema debe mostrar la opción seleccionada y estar listo para ejecutar la maximización del grafo.			
<u>02-09</u>	Presiona el botón "Resolver" para aplicar el algoritmo de Kruskal en modo de maximización.		Sobre el grafo en pantalla, se debe resaltar el camino óptimo (árbol de expansión máxima) con colores diferenciados.			
<u>02-10</u>	Verifica que el valor de la suma de los pesos de la ruta óptima se muestre en pantalla.		El sistema debe mostrar el valor de la suma total de la ruta óptima calculada.			
<u>TC-KS-01</u>	Verificar que el usuario pueda aplicar el algoritmo de Kruskal en el editor de grafos, seleccionando la opción de	High	- Backend ejecutado en un		<u>TC-KS-02.png</u>	Passed

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
	minimizar para obtener el árbol de expansión mínimo y visualizar tanto la ruta óptima resaltada como el valor de la suma total de la ruta.		contenedor. -Frontend del proyecto ejecutado en local. -. Acceso al proyecto a través del navegador - Tener una cuenta registrada en la plataforma. - Iniciar sesión con la cuenta registrada.			
<u>01-01</u>	Accede a la aplicación en http://localhost:5173 o en el dominio desplegado.			La aplicación debe abrirse en la página de inicio (home).		
<u>01-02</u>	Dirígete al apartado de "Grafos" desde el navbar o accediendo a http://localhost:5173/graphs_home .			Se debe mostrar una lista de todas las opciones de grafos disponibles, junto con una breve descripción de cada una.		
<u>01-03</u>	Selecciona la opción "Grafos".			La aplicación debe redirigir al editor de grafos, mostrando la opción de abrir el editor y una breve descripción del módulo de grafos.		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>01-04</u>	Selecciona la opción "Ir a Grafos".			El editor de grafos debe mostrarse en pantalla, con el menú de edición de grafos habilitado para el usuario.		
<u>01-05</u>	En el menú del editor, selecciona la opción "Kruskal".			Debe aparecer una descripción del algoritmo de Kruskal con un botón de "Ir a Kruskal".		
<u>01-06</u>	Presiona "Ir a Kruskal" para acceder a la funcionalidad del algoritmo de Kruskal.			El editor debe cargar con opciones para aplicar el algoritmo de Kruskal.		
<u>01-07</u>	Crea un grafo básico en el editor, siguiendo los pasos del test case TC-JAC-M03-001 como referencia.			El grafo debe visualizarse correctamente en el editor con todos los nodos y aristas creados.		
<u>01-08</u>	En el menú de Kruskal, selecciona la opción "Minimizar" en el check de opciones.			El sistema debe mostrar la opción seleccionada y estar listo para ejecutar la minimización del grafo.		
<u>01-09</u>	Presiona el botón "Resolver" para aplicar el algoritmo de Kruskal en modo de minimización.			Sobre el grafo en pantalla, se debe resaltar el camino óptimo (árbol de expansión mínimo)		

Aa Codigo	Descripción	Severidad	Precondiciones	Resultado esperado	Evidencias	Estado ejecución
				con colores diferenciados.		
<u>01-10</u>	Verifica que el valor de la suma de los pesos de la ruta óptima se muestre en pantalla.			El sistema debe mostrar el valor de la suma total de la ruta óptima calculada.		
<u>TC-JH- 12</u>	Verificar que los archivos de grafos de Johnson se puedan guardar de manera local y virtual	Low	- Backend corriendo - Frontend corriendo - Se accedió correctamente por login - Se accedió correctamente hasta la vista de algoritmos - Se accedió correctamente hasta la vista de Johnson - MiniO corriendo - Tengo un grafo cualquiera		<u>JohnsonCorrectTC12.png</u>	Passed
<u>12-01</u>	Presiona el botón 			Aparece un prompt invitando a colocar el nombre		Passed
<u>12-02</u>	Escribe el nombre Johnson			Se descarga un archivo Johnson.json		Passed

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>12-03</u>	En la vista de selección de archivo selecciono [upload] y puedo cargar el archivo Johnson.json			El archivo aparece en la pantalla de selección		Passed
<u>12-04</u>	Selecciono el archivo recién cargado			Se abre mi archivo en el editor		Passed
<u>TC-JH-11</u>	Verificar que la vista de algoritmos de Johnson permita construir un grafo de actividades con las actividades en las aristas	High	<ul style="list-style-type: none"> - Backend corriendo - Frontend corriendo - Se accedió correctamente por login - Se accedió correctamente hasta la vista de algoritmos - Se accedió correctamente hasta la vista de Johnson 		<u>JohnsonCorrectTC11.png</u>	Passed
<u>11-01</u>	Seleccionar el botón + y hacer clic sobre el lienzo			Aparece un nodo con nombre "Nodo1" en la posición clickeada		Passed
<u>11-02</u>	Hacer clic sobre el lienzo			Aparece un nodo con nombre "Nodo2" en la posición clickeada		Passed
<u>11-03</u>	Seleccionar el botón 📌 y seleccionar los últimos dos nodos creados manteniendo shift durante el proceso			Aparece una arista desde el primer nodo clickeado con 0 como valor		Passed

Aa Codigo	Descripción	Severidad	Precondiciones	Resultado esperado	Evidencias	Estado ejecución
11-04	Repetir los pasos 1 a 3 hasta tener un grafo similar a la referencia			(ver evidencia anexa)		Passed
11-05	Seleccionar una de las aristas y hacer clic sobre el botón de edición			Aparece un modal que indica el nuevo valor que se le asignara al arista		Passed
11-06	Ingresar un valor numerico y guardar cambios			El modal se cierra y la arista ahora tiene ese valor		Passed
11-07	Repetir los pasos 11-01 a 11-06 hasta construir un grafo direccionado con un nodo inicial y un nodo final sin generar bucles			Ver evidencia		Passed
TC-GF-03	Verificar que el sistema permite guardar la estructura y configuraciones del grafo en un archivo JSON descargable y que permite cargar dicho archivo para restaurar el grafo sin pérdida de información ni de la estructura inicial.	Medium	<ul style="list-style-type: none"> - Backend ejecutado en un contenedor. -Frontend del proyecto ejecutado en local. - Acceso al proyecto a través del navegador - Tener una cuenta registrada en la plataforma. - Iniciar sesión con la cuenta registrada. 		TC-GF-03.png	Passed
03-01	Accede a la aplicación en http://localhost:5173 o en el dominio desplegado.			La aplicación debe abrirse en la página		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				de inicio (home).		
<u>03-02</u>	Dirígete al apartado de "Grafos" desde el navbar o accediendo a http://localhost:5173/graphs_home .			Se debe mostrar una lista de todas las opciones de grafos disponibles, junto con una breve descripción de cada una.		
<u>03-03</u>	Selecciona la opción "Grafos".			La aplicación debe redirigir al editor de grafos, mostrando la opción de abrir el editor y una breve descripción del módulo de grafos.		
<u>03-04</u>	Selecciona la opción "Ir a Grafos".			Se debe abrir una ventana donde el usuario pueda elegir entre agregar un nuevo archivo relacionado a este módulo o crear un grafo nuevo desde cero.		
<u>03-05</u>	Selecciona la opción "Crear".			El editor de grafos debe mostrarse en pantalla, con el menú de edición de grafos habilitado para el usuario.		
<u>03-06</u>	Agrega varios nodos y aristas para estructurar un grafo básico.			Los nodos y aristas deben aparecer en		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	⚙ Resultado esperado	📎 Evidencias	⌵ Estado ejecución
				el editor de grafos según las configuraciones y posiciones establecidas.		
03-07	Haz clic en el botón "Guardar" desde el menú de opciones del editor de grafos.			Se debe descargar automáticamente un archivo en formato JSON que contenga la estructura del grafo, nodos, aristas y configuraciones actuales.		
03-08	Restablece la instancia del navegador (cerrar y volver a abrir o reiniciar la página).			La aplicación debe volver a abrirse en el estado inicial, sin ningún grafo cargado en el editor.		
03-09	Regresa al editor de grafos y selecciona la opción de "Cargar" en el menú de opciones.			Debe abrirse un cuadro de diálogo que permita seleccionar un archivo JSON para cargar.		
03-10	Carga el archivo JSON previamente descargado.			El editor debe reconstruir el grafo en el mismo estado en que fue guardado, con todos los nodos, aristas y configuraciones en		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				sus posiciones originales.		
<u>03-11</u>	Modifica el grafo agregando un nodo y una arista adicionales.			Las modificaciones deben reflejar visualmente en el editor.		
<u>03-12</u>	Guarda nuevamente el grafo modificado en un archivo JSON.			Se debe descargar un nuevo archivo JSON que refleje los cambios realizados, incluyendo el nuevo nodo y arista.		
<u>03-13</u>	Carga el nuevo archivo JSON modificado.			El grafo debe reconstruirse en el editor, mostrando todos los nodos y aristas según el último estado guardado.		
<u>TC-GF-02</u>	Verificar que el sistema permite al usuario visualizar una matriz de adyacencia precisa de acuerdo con la estructura actual del grafo y que esta matriz se actualiza en tiempo real cuando se agregan o modifican nodos y aristas.	High	- Backend ejecutado en un contenedor. -Frontend del proyecto ejecutado en local. - Acceso al proyecto a través del navegador - Tener una cuenta		<u>BGJH-solved.png</u>	Passed

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
			registrada en la plataforma. - Iniciar sesión con la cuenta registrada.			
<u>02-01</u>	Accede a la aplicación en http://localhost:5173 o el dominio desplegado.			La aplicación debe abrirse en la página de inicio (home).		
<u>02-02</u>	Dirígete al apartado de "Grafos" desde el navbar o accediendo a http://localhost:5173/graphs_home .			Se debe mostrar una lista de todas las opciones de grafos disponibles, junto con una breve descripción de cada una.		
<u>02-03</u>	Selecciona la opción "Grafos".			La aplicación debe redirigir al editor de grafos, mostrando la opción de abrir el editor y una breve descripción del módulo de grafos.		
<u>02-04</u>	Selecciona la opción "Ir a Grafos".			Se debe abrir una ventana donde el usuario pueda elegir entre agregar un nuevo archivo relacionado a este módulo o crear un grafo nuevo desde cero.		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>02-05</u>	Selecciona la opción "Crear".			El editor de grafos debe mostrarse en pantalla, con el menú de edición de grafos habilitado para el usuario.		
<u>02-06</u>	Haz clic en el botón "Matriz de Adyacencia" antes de crear nodos y aristas.			Un mensaje debe indicar que no es posible generar la matriz de adyacencia porque no hay nodos creados en el grafo.		
<u>02-07</u>	Selecciona la opción "Crear Nodo" en el menú inferior y coloca varios nodos en el editor.			Los nodos deben aparecer en las ubicaciones seleccionadas, visualmente representados en el editor.		
<u>02-08</u>	Selecciona al menos dos nodos y utiliza la opción para agregar una arista entre ellos.			Debe aparecer una arista que conecte los nodos seleccionados, mostrando visualmente la relación entre ellos.		
<u>02-09</u>	Haz clic nuevamente en el botón "Matriz de Adyacencia" después de crear nodos y aristas.			Debe aparecer un pop-up con la matriz de adyacencia, representando las conexiones actuales		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				entre los nodos del grafo.		
<u>02-10</u>	Modifica el grafo agregando un nodo adicional y una arista conectándolo a uno de los nodos existentes.			El nuevo nodo y la arista deben aparecer en el editor, mostrando correctamente la nueva relación establecida.		
<u>02-11</u>	Haz clic nuevamente en "Matriz de Adyacencia" para actualizar la matriz.			El pop-up de la matriz de adyacencia debe reflejar los cambios realizados en el grafo, incluyendo el nuevo nodo y la arista añadidos.		
<u>TC-GF-01</u>	Verificar la funcionalidad de creación de nodos y aristas en el módulo de grafos, asegurando que el editor permite la interacción básica para construir grafos.	High	<ul style="list-style-type: none"> - Backend ejecutado en un contenedor. -Frontend del proyecto ejecutado en local. - Acceso al proyecto a través del navegador - Tener una cuenta registrada en la plataforma. - Iniciar sesión 		<u>TC-GF-01.png</u>	Passed

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
			con la cuenta registrada.			
<u>01-01</u>	Accede a la aplicación en http://localhost:5173 o el dominio desplegado.			La aplicación debe abrirse en la página de inicio (home).		
<u>01-02</u>	Dirígete al apartado de "Grafos" desde el navbar o accediendo a http://localhost:5173/graphs_home .			Se debe mostrar una lista de todas las opciones de grafos disponibles, junto con una breve descripción de cada una.		
<u>01-03</u>	Selecciona la opción "Grafos".			La aplicación debe redirigir al editor de grafos, mostrando la opción de abrir el editor y una breve descripción del módulo de grafos.		
<u>01-04</u>	Selecciona la opción "Ir a Grafos".			Se debe abrir una ventana donde el usuario pueda elegir entre agregar un nuevo archivo relacionado a este módulo o crear un grafo nuevo desde cero.		
<u>01-05</u>	Selecciona la opción "Crear".			El editor de grafos debe mostrarse en pantalla, con el menú de edición de		

Aa Codigo	Descripción	Severidad	Precondiciones	Resultado esperado	Evidencias	Estado ejecución
				grafos habilitado para el usuario.		
<u>01-06</u>	Selecciona la opción "Crear Nodo" en el menú inferior. Arrastra y coloca nodos en el editor en la posición deseada.			Los nodos deben aparecer en las ubicaciones seleccionadas, visualmente representados en el editor.		
<u>01-07</u>	Selecciona al menos dos nodos y utiliza la opción para agregar una arista entre ellos.			Debe aparecer una arista que conecte los dos nodos seleccionados, representando correctamente la relación entre ellos.		
<u>TC-DJ-02</u>	Verificar que la interfaz maneja adecuadamente errores al intentar procesar datos no válidos en Dijkstra.	Medium	- Sistema corriendo - Grafo con nodos o arcos no válidos o datos incompletos - Debe haberse iniciado sesión como usuario del sistema.		<u>DijkstraCorrectTCDJ02.png</u>	Passed
<u>02-01</u>	Ingresar datos no válidos en el grafo (por ejemplo, peso no numérico o arcos incompletos).			La interfaz debe mostrar un mensaje de error claro y específico.		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>02-02</u>	Intentar ejecutar Dijkstra con los datos no válidos.			Debe detener la ejecución y mostrar un mensaje sin impactar el rendimiento.		
<u>02-03</u>	Corregir los datos e intentar nuevamente.			La interfaz debe permitir ejecutar los cambios sin errores.		
<u>02-04</u>	Validar que se calcule correctamente.			Debe devolver el camino más corto y el peso total correcto entre nodos de inicio y fin.		
<u>TC-DJ-01</u>	Verificar que el algoritmo de Dijkstra calcula correctamente el camino mínimo en un grafo simple con datos válidos.	Medium	- Sistema corriendo - Grafo simple con nodos y arcos con pesos válidos - Inicio de sesión como usuario.		DijkstraCorrectTCDJ01.jpg	Passed
<u>01-01</u>	Ejecutar el algoritmo de Dijkstra en el grafo.			Debe devolver el camino más corto y el peso total correcto entre nodos de inicio y fin.		
<u>01-02</u>	Verificar que los caminos intermedios respeten el peso mínimo en cada paso.			Debe mostrar cada paso correctamente y solo seleccionar arcos de menor peso.		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>01-03</u>	Llenar los campos del formulario con datos válidos			Deben haber validaciones del tipo de dato que se ingresa		
<u>01-04</u>	Validar que se calcule correctamente cuando no hay camino entre los nodos.			Debe mostrar un mensaje o valor de "Sin conexión" si no hay camino posible.		
<u>TC-DB-02</u>	Verificar que cada algoritmo tenga un texto de ayuda o "help" visible y comprensible que explique su funcionamiento y pasos de uso.	Low	- Sistema corriendo - Usuario autenticado y en el dashboard donde están disponibles los algoritmos.		<u>DashboardCorrectTCDB02.png</u>	Passed
<u>02-01</u>	Localizar el texto de ayuda asociado al algoritmo de Dijkstra en el dashboard.			El texto de ayuda debe estar claramente visible y explicar en términos sencillos el funcionamiento del algoritmo.		
<u>02-02</u>	Revisar que el texto de ayuda incluya ejemplos o pasos básicos para el uso de Dijkstra.			El texto de ayuda debe incluir una breve descripción y pasos clave para utilizar el algoritmo correctamente.		
<u>02-03</u>	Localizar el texto de ayuda asociado al algoritmo de Compet en el dashboard.			El texto de ayuda para Compet debe cumplir con los		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	⌚ Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				mismos requisitos, brindando una guía clara y ejemplos de uso.		
<u>02-04</u>	Revisar que el texto de ayuda incluya ejemplos o pasos básicos para el uso de Compet.			El texto de ayuda debe incluir una breve descripción y pasos clave para utilizar el algoritmo correctamente.		
<u>TC-DB- 01</u>	Verificar que el dashboard, en los algoritmos de Dijkstra y Compet, refleja correctamente los cambios cuando se modifica un nodo o arco en el grafo.	High	- Sistema corriendo - Grafo cargado con al menos dos nodos y un arco entre ellos visible en el dashboard - Debe haberse iniciado sesión como usuario.		<u>DashboardCorrectTCDB01.jpg</u>	Passed
<u>01-01</u>	Seleccionar un arco existente en el dashboard.			Debe destacarse visualmente el arco seleccionado.		
<u>01-02</u>	Cambiar el peso o la dirección del arco.			El cambio debe guardarse sin errores y reflejarse de inmediato en el dashboard.		
<u>01-03</u>	Ejecutar el algoritmo Compet o Dijkstra después de la modificación.			El algoritmo debe considerar el cambio en el cálculo y		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				actualizar el resultado en el dashboard.		
<u>01-04</u>	Verificar que el resultado refleje el nuevo peso o dirección del arco modificado.			El resultado debe actualizarse y mostrar el impacto del cambio en los cálculos de manera precisa.		
<u>TC-CP-01</u>	Verificar que el algoritmo de Compet procesa correctamente nodos con valores extremos.	Medium	- Sistema corriendo - Grafo con valores altos en algunos nodos - Debe haberse iniciado sesión como usuario del sistema		<u>CompetCorrectCP01.png</u>	Passed
<u>01-01</u>	Cargar un grafo con pesos negativos y extremos (p.ej., 99999).			Debe aceptar todos los valores sin errores de ejecución.		
<u>01-02</u>	Ejecutar el algoritmo Compet con los datos cargados			Debe calcular correctamente sin errores y sin ciclos infinitos.		
<u>01-03</u>	Verificar que los resultados reflejan los pesos negativos y extremos.			El resultado debe incluir los pesos negativos y valores altos sin errores.		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>TC-BT-02</u>	Verificar que el usuario pueda guardar una estructura de árbol binario y cargarla en futuras sesiones.	Medium	El usuario debe tener acceso al módulo de Árboles Binarios.		<u>Loom Evidencias</u>	Passed
<u>02-01</u>	El usuario construye un árbol binario ingresando valores.			El árbol binario se construye correctamente con los valores ingresados por el usuario, mostrando una estructura visual clara y precisa en la pantalla.		
<u>02-02</u>	El usuario guarda la estructura del árbol.			La estructura del árbol se guarda exitosamente, y el sistema confirma que el archivo ha sido creado sin errores.		
<u>02-03</u>	El usuario cierra la sesión o cierra el módulo.			Al cerrar la sesión o el módulo, no se pierden datos ni configuraciones, y el sistema finaliza correctamente las operaciones en curso.		
<u>02-04</u>	El usuario inicia sesión nuevamente y carga la estructura guardada.			Al iniciar sesión nuevamente y cargar la estructura		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				guardada, el árbol se reconstruye exactamente como se guardó, incluyendo todos los nodos, conexiones y valores previamente definidos.		
<u>TC-BT-01</u>	Verificar que el usuario pueda ajustar la visualización del árbol binario centrándolo y ajustando el zoom para ver la estructura completa.	Low	- El usuario debe tener un árbol binario creado y visible en la pantalla.		<u>Loom Evidencias</u>	Passed
<u>01-01</u>	El usuario selecciona la opción para centrar el árbol en la vista.			Al seleccionar la opción para centrar el árbol, este se posiciona automáticamente en el área visible de la pantalla, ajustándose al centro sin errores.		
<u>01-02</u>	El usuario observa que el árbol se centra en el área visible de la pantalla.			El árbol aparece centrado en la vista, con todos sus nodos y conexiones visibles, sin que ningún elemento quede fuera del área visible.		
<u>01-03</u>	El usuario utiliza las opciones de zoom (acercar y alejar) para ajustar la visualización del árbol.			Las opciones de zoom (acercar y		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	⚙ Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				alejarse) responden correctamente, permitiendo al usuario ajustar la visualización sin interrupciones ni retrasos.		
<u>01-04</u>	El usuario observa cómo cambia el tamaño del árbol en la vista.			El tamaño del árbol cambia proporcionalmente en la vista, mostrando todos los elementos de manera clara y sin distorsiones.		
<u>TC-AG-05</u>	Verificar que la vista del Algoritmo de Asignación permite abrir un dashboard guardado en el dispositivo	Medium	<ul style="list-style-type: none"> - Todos los servicios necesarios (Docker, backend, frontend) deben estar operativos. - El usuario debe estar registrado y haber iniciado sesión. - Existencia de un grafo configurado 		<u>TCAG05.png</u>	Passed
<u>05-01</u>	Limpiar el dashboard del Algoritmo de Asignación si existiera algún grafo			El dashboard se encuentra vacío		

Aa Codigo	Descripción	▲ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	📌 Estado ejecución
<u>05-02</u>	Seleccionar el icono que señala "Abrir el Dashboard".			Se abre el navegador, para seleccionar el archivo a abrir		
<u>05-03</u>	Seleccionar un archivo que se permita en el sistema			Se selecciona un archivo y esté es abierto por el sistema.		
<u>05-04</u>	Visualizar el diseño del grafo abierto en el dashboard			El dashboard se encuentra con el grafo del archivo abierto		
<u>TC-AG-04</u>	Verificar que las configuraciones realizadas en el dashboard de Algoritmo del Asignación se guarden en el dispositivo.	Medium	<ul style="list-style-type: none"> - Todos los servicios necesarios (Docker, backend, frontend) deben estar operativos. - El usuario debe estar registrado y haber iniciado sesión. - Existencia de un grafo configurado 		<u>TCAG04.png</u>	Passed
<u>04-01</u>	Seleccionar el icono que señala "Guardar el Dashboard".			Se presenta una interfaz o confirmación para		

Aa Codigo	Descripción	Severidad	Precondiciones	Resultado esperado	Evidencias	Estado ejecución
				guardar el dashboard.		
<u>04-02</u>	Ingresar nombre del archivo a guardar.			Se abrió el explorador de archivos		
<u>04-03</u>	Ingresar el nombre en el explorador de archivos y seleccionar su ruta de almacenamiento.			El dashboard se guarda correctamente y el sistema confirma que el estado se ha guardado.		
<u>04-04</u>	Verificar que el archivo se encuentre en la ruta elegida con el nombre registrado.			El archivo json se encuentra en la ruta elegida.		
<u>TC-AG-03</u>	Verificar que el algoritmo de Asignación calcule el resultado óptimo, minimizando los pesos de las aristas	High	<div>- Todos los servicios necesarios (Docker, backend, frontend) deben estar operativos.</div> <div>- El usuario debe estar registrado y haber iniciado sesión.</div> <div>- Existencia de un grafo configurado</div>		<u>TCAG03.png</u>	Passed
<u>03-03</u>	Visualizar y validar los resultados mostrados en la interfaz.			Los resultados del algoritmo se		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🔄 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
				muestran claramente en la interfaz y presenta valores válidos.		
<u>03-02</u>	Ejecutar el algoritmo de asignación.			El algoritmo se ejecuta sin errores y comienza el proceso de asignación.		
<u>03-01</u>	Seleccionar "Minimizar" la asignación			Muestra la elección de asignación y muestra una opción para ejecutarse		
<u>TC-AG-02</u>	Verificar que el algoritmo de Asignación calcule el resultado óptimo, maximizando los pesos de las aristas	High	<ul style="list-style-type: none">- Todos los servicios necesarios (Docker, backend, frontend) deben estar operativos.- El usuario debe estar registrado y haber iniciado sesión.- Existencia de un grafo configurado		<u>TCAG02.png</u>	Passed
<u>02-01</u>	Seleccionar "Maximizar" la asignación			Muestra la elección de asignación y muestra una opción para ejecutarse		

Aa Codigo	≡ Descripción	⚠ Severidad	→ Precondiciones	🕒 Resultado esperado	📎 Evidencias	⬇ Estado ejecución
<u>02-02</u>	Ejecutar el algoritmo de asignación.			El algoritmo se ejecuta sin errores y comienza el proceso de asignación.		
<u>02-03</u>	Visualizar y validar los resultados mostrados en la interfaz.			Los resultados del algoritmo se muestran claramente en la interfaz y presenta valores válidos.		
<u>TC-AG-01</u>	Verificar que el algoritmo de Asignación no se ejecute si el usuario no está autenticado	Medium	<ul style="list-style-type: none">- Todos los servicios necesarios (Docker, backend, frontend) deben estar operativos.- El usuario debe estar registrado, pero su sesión debe haber caducado.- Existencia de un grafo configurado		<u>TCAG01.png</u>	Passed
<u>01-01</u>	Seleccionar si "Maximizar" o "Minimizar" la asignación			Muestra la elección de asignación y muestra una opción para ejecutarse		

Aa Codigo	≡ Descripción	▲ Severidad	→ Precondiciones	⌂ Resultado esperado	📄 Evidencias	⌵ Estado ejecución
<u>01-02</u>	Ejecutar el algoritmo de asignación.			El algoritmo se ejecuta sin errores y comienza el proceso de asignación.		
<u>01-03</u>	Verificar que no se muestre ningún resultado.			Los resultados del algoritmo NO se muestran en la interfaz.		

Anexo B - Bugs identificados

Aa Codigo	≡ Descripción	➡ Resultado esperado	⦿ Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
<u>BG-AG-04</u>	Cuando se quiere resolver la Asignación con costos en los pesos de los arcos muy elevados. La interfaz de la matriz de adyacencia se desborda.	La interfaz debe manejar adecuadamente la visualización de grandes valores sin desbordamiento.	La interfaz se desborda al ingresar valores grandes.	Low	<u>BGAG04.png</u>	@2024/10/27
<u>04-01</u>	Crear más de 2 nodos					
<u>04-02</u>	Ingresar pesos con valores grandes en las aristas de los nodos.					
<u>04-03</u>	Visualizar la Matriz de Adyacencia					
<u>BG-AT-02</u>	La autenticación con Facebook no está funcionando. Tanto en las secciones de login como de registro, al presionar el botón de autenticación con Facebook, no se redirige a ninguna página ni se	La autenticación con Face debería de ser funcional y en caso de ser seleccionada y exitosa generar token de acceso	El botón de autenticación con google simplemente no lleva a ningún lado	Low	<u>IS-JAC-002.mov</u>	@2024/10/28

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	muestra el menú de acceso con Facebook.					
<u>02-01</u>	Accede a la aplicación en http://localhost:5173 o en el dominio desplegado.					
<u>02-02</u>	Dirígete a la sección de "Login" desde el menú de navegación o página de inicio.					
<u>02-03</u>	En la sección de Login, selecciona la opción "Iniciar sesión con Facebook".					
<u>02-04</u>	Dirígete a la sección de "Registro" desde el menú de navegación o página de inicio.					

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
<u>02-05</u>	En la sección de Registro, selecciona la opción "Registrar con Facebook".					
<u>BG- CW-01</u>	Al acceder al apartado del teleférico y abrir el formulario para la optimización de rutas, el dropdown que debería permitir encoger el apartado de las tarjetas no responde a la interacción del usuario.	Que el apartado de tarjetas se encoja al hacer click en el dropdown.	Al hacer click en el dropdown no ocurre la interacción deseada..	Low	<u>Telef01.png</u>	@2024/10/30
<u>01-01</u>	El usuario accede al apartado del teleférico.					
<u>01-02</u>	El usuario abre el formulario de optimización de rutas.					
<u>01-03</u>	El usuario intenta utilizar el dropdown para encoger el apartado de las tarjetas.					
<u>BG-DB- 01</u>	Al editar el peso de un arco o el nombre de un nodo en el dashboard, los campos	El campo de edición debería mostrar el valor actual del peso	El campo de edición aparece vacío, sin mostrar	Low	<u>BGDB01.png</u>	@2024/10/15

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	de edición no muestran el valor anterior, impidiendo una referencia visual del valor actual.	del arco o nombre del nodo, permitiendo que el usuario vea y actualice el valor de manera directa.	el valor actual del peso del arco o el nombre del nodo.			
<u>01-01</u>	Seleccionar un arco o nodo en el dashboard.					
<u>01-02</u>	Elegir la opción de editar el peso del arco o el nombre del nodo.					
<u>01-03</u>	Observar los campos de entrada para editar el valor.					
<u>BG-DB-02</u>	Cuando se accede al dashboard de cualquier algoritmo y se despliega la barra lateral para seleccionar la opción "Ir a inicio", la navegación se completa correctamente. Sin embargo, al intentar regresar a la sección "Grafos" desde el navbar en la nueva pantalla, la página se queda estática y no	Debería existir una opción para quitar la ruta marcada, permitiendo editar libremente todos los arcos del grafo después de resolver el algoritmo de Dijkstra.	No se puede editar ningún arco que forma parte de la ruta resaltada, y no hay una opción para quitar la marca de la ruta, lo que limita la manipulación del grafo.	Low	BGDB02.png	@2024/12/14

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	permite hacer scroll, lo que impide acceder a los algoritmos y otras opciones en esa sección.					
<u>02-01</u>	Acceder al dashboard de cualquier algoritmo.					
<u>02-02</u>	Desplegar el sidebar y seleccionar "Ir a inicio".					
<u>02-03</u>	En la nueva pantalla, seleccionar "Grafos" desde el navbar.					
<u>02-04</u>	Intentar hacer scroll en la pantalla de "Grafos".					
<u>BG-DJ-01</u>	Al ejecutar el algoritmo de dijkstra con un grafo no válido (debido a que no tiene los arcos completos) el error que se muestra es poco claro.	El mensaje de error debería mostrar que el error específico es que no se introdujo en grafo válido para el algoritmo.	El mensaje de error está en inglés, indicando que hubo un error al ejecutar el algoritmo	Low	<u>BGDJ01.png</u>	@2024/10/15
<u>01-01</u>	Insertar 2 nodos en el dashboard del algoritmo Dijkstra.					
<u>01-02</u>	Seleccionar como nodo de inicio al nodo 1, y como					

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	nodo de destino al nodo 2.					
<u>01-03</u>	Presionar el botón "Resolver"					
<u>BG-GH-01</u>	En la pantalla de grafos, los botones diseñados para redirigir a apartados específicos (por ejemplo, "Árboles Binarios") no funcionan correctamente.	Al hacer clic en los botones, estos se quedan en la posición actual.	Al hacer clic en los botones, no ocurre ningún desplazamiento hacia el apartado deseado en la página, lo que dificulta la navegación dentro del módulo.	Low	<u>Loom Evidencias</u>	@2024/11/15
<u>01-01</u>	El usuario accede a la pantalla de grafos.					
<u>01-02</u>	El usuario hace clic en un botón, por ejemplo, "Árboles Binarios".					
<u>01-03</u>	El usuario intenta hacer clic en otros botones para verificar el comportamiento.					

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
<u>BG-JH-01</u>	Error al mover nodos con aristas loop a sí mismos	La arista y su valor se deberían mover junto con el nodo	Cuando se crea una arista conectada a si misma, su peso no se mueve junto al elemento asociado	Low	<u>JohnsonBug01.png</u>	@2024/03/15
<u>01-01</u>	Se selecciona un nodo y se presiona el botón 					
<u>01-02</u>	Arrastrar el nodo hasta una posición diferente					
<u>BG-AG-01</u>	Cuando se crea arcos de un nodo a otro. Se puede crear múltiples arcos del mismo nodo origen al mismo nodo destino.	No se debe permitir más de un arco en la misma dirección entre dos nodos.	Se permite crear múltiples aristas en la misma dirección.	Medium	<u>BGAG01.png</u>	@2024/12/14
<u>01-01</u>	Crear dos nodos, A y B.					
<u>01-02</u>	Conectar A con B con un arco.					
<u>01-03</u>	Intentar conectar A con B con otro arco.					
<u>03-03</u>	Seleccionar "Resolver Asignación"					

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
<u>BG-AG-03</u>	Cuando se intenta calcular la solución del Algoritmo de Asignación con un grafo incorrecto, no se valida la estructura del grafo que debe existir para poder realizar la asignación	Debería mostrar un mensaje de alerta que indique la estructura qu debe tener el grafo para poder aplicar asignación	Calcula un solución inadecuada,	Medium	<u>BGAG03.png</u>	@2024/10/27
<u>03-01</u>	Tener un Dashboard con un grafo inadecuado, incluyendo bucles y arcos seguidos entre nodos					
<u>03-02</u>	Seleccionar si "Maximizar" o "Minimizar" la solución					
<u>03-04</u>	Observar como la solución presenta un resultado inadecuado.					
<u>BG-AT-01</u>	La autenticación con Google no está funcionando. Tanto en las secciones de login como de registro, al presionar el botón de autenticación con Google, no se redirige a ninguna página ni se	La autenticación con Google debería de ser funcional y en caso de ser seleccionada y exitosa generar token de acceso	El botón de autenticación con google simplemente no lleva a ningún lado	Medium	<u>IS-JAC-001.mov</u>	@2024/10/28

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	muestra el menú de acceso con Google.					
<u>01-01</u>	Accede a la aplicación en http://localhost:5173 o en el dominio desplegado.					
<u>01-02</u>	Dirígete a la sección de "Login" desde el menú de navegación o página de inicio.					
<u>01-03</u>	En la sección de Login, selecciona la opción "Iniciar sesión con Google".					
<u>01-04</u>	Dirígete a la sección de "Registro" desde el menú de navegación o página de inicio.					
<u>01-05</u>	En la sección de Registro, selecciona la opción "Registrar con Google".					
<u>BG-DJ-03</u>	Después de ejecutar el algoritmo de Dijkstra, el sistema marca el camino más corto en el grafo resaltando los arcos y nodos de la ruta. Sin	Debería existir una opción para quitar la ruta marcada, permitiendo editar libremente todos los arcos del grafo	No se puede editar ningún arco que forma parte de la ruta resaltada, y no hay una opción para quitar la	Medium	BGDJ03.png	@2024/10/28

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	embargo, los arcos dentro de la ruta marcada no pueden editarse, lo que impide realizar modificaciones en el grafo mientras esté visible la ruta. Además, no existe una opción para quitar esta ruta marcada, limitando la funcionalidad del grafo.	después de resolver el algoritmo de Dijkstra.	marca de la ruta, lo que limita la manipulación del grafo.			
<u>03-01</u>	Insertar un grafo en el dashboard dijkstra.					
<u>03-02</u>	Ejecutar el algoritmo de Dijkstra para calcular el camino más corto.					
<u>03-03</u>	Intentar editar los arcos que forman parte de la ruta resaltada.					
<u>BG-JH-02</u>	Error de grafos inválidos, vacíos y de conexiones incorrectas	Aparece un mensaje de error indicando por qué el grafo es inválido	Cuando se intenta solucionar el algoritmo de Johnson con un grafo incorrecto no sucede nada	Medium	<u>JohnsonBug02.png</u>	@2024/10/15

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
<u>02-01</u>	Construir un grafo según la evidencia					
<u>02-02</u>	Desconectar el grafo para que quede en dos piezas					
<u>02-03</u>	Intentar resolver el algoritmo de ruta crítica					
<u>BG-JH-03</u>	Error en grafos con un loop	La lógica debería identificar los bucles de antemano y responder con un mensaje de error apropiado	Cuando se intenta solucionar un grafo con un bucle, el algoritmo entra en un loop infinito del que nunca sale	Medium	<u>JohnsonBug03.png</u>	@2024/10/15
<u>03-01</u>	Crear un grafo cualquiera que sea valido					
<u>03-02</u>	Conectar el último nodo al primero, creando un bucle					
<u>03-03</u>	Intentar resolver el algoritmo de ruta crítica					
<u>BG-JH-04</u>	Error en grafos con más de una entrada o salida	Cuando se intenta resolver un grafo de esta forma debería mostrarse un mensaje de error	Cuando se intenta solucionar un grafo con más de una entrada o salida, la ruta	Medium	<u>JohnsonBug04.png</u>	@2024/10/15

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
		indicando la representación invalida	crítica aparece incorrecta y mezclada			
<u>04-01</u>	Crear un grafo de acuerdo con la referencia					
<u>04-02</u>	Intentar resolver el grafo					
<u>BG- NW-04</u>	El sistema permite crear conexiones con datos inconsistentes sin notificar al usuario sobre los errores en los datos ingresados.	El sistema debe notificar al usuario sobre la inconsistencia en la entrada de datos, previniendo errores en la creación del grafo.	El sistema evita la inconsistencia de los datos, pero no notifica sobre errores.	Medium	<u>Captura de pantalla 2024-10-28 a la(s) 23.02.56.png, Captura de pantalla 2024-10- 28 a la(s) 23.02.33.png</u>	@2024/10/17
<u>04-01</u>	Definir un nodo en el editor.					
<u>04-02</u>	Ingresar datos válidos en todos los campos y enviar el formulario.					
<u>04-03</u>	Intentar crear una conexión con datos inconsistentes (por ejemplo, un peso no numérico).					
<u>BG-ST- 02</u>	En el módulo de Sorts, al generar una lista de	El sistema ordena la lista y presenta los	Los números ordenados se	Medium	<u>TCST03.png</u>	@2024/10/28

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	números aleatorios superior a 50 elementos, el sistema presenta problemas de visualización al ejecutar el ordenamiento. Los datos se muestran incompletos o superpuestos en pantalla, lo que afecta la visualización adecuada del resultado.	resultados de forma clara y legible en pantalla.	muestran de forma incompleta y superpuestos en la pantalla.			
<u>02-01</u>	El usuario accede al módulo de Sorts.					
<u>02-02</u>	El usuario selecciona la opción para generar números aleatorios.					
<u>02-03</u>	El usuario ingresa un valor superior a 50 (por ejemplo, 75) y genera la lista.					
<u>02-04</u>	El usuario selecciona un algoritmo de ordenamiento (por ejemplo, Selection Sort).					
<u>BG-AG-02</u>	En algunos casos, cuando el dashboard está vacío, se muestran resultados de	No se deben mostrar datos si el dashboard está vacío.	Se muestran resultados de valores anteriores	High	<u>BGAG02.png</u>	@2024/10/27

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	valores anteriores en la matriz de adyacencia.					
<u>02-01</u>	Tener un Dashboard con un grafo					
<u>02-02</u>	Salir del Dashboard, sn guardarlo					
<u>02-03</u>	Volver a navegar al dashboard.					
<u>02-04</u>	Verificar que el dashboard esté vacío.					
<u>02-05</u>	Observar la visualización de la matriz de adyacencia con valores de un grafo anterior.					
<u>BG-AG-05</u>	Al resolver un algoritmo, si la sesión del token ha expirado y este ya no es válido, el sistema no notifica al usuario que debe volver a iniciar sesión. Además, el usuario sigue teniendo acceso para resolver los algoritmos, pero estos ya no	El sistema debe redirigir al usuario para reautenticación cuando el token expira.	No ocurre redirección ni refresco del token, el sistema sigue funcionando y no se puede realizar acciones que manden solicitud al backend.	High	<u>BGAG05.png</u>	@2024/10/27

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	proporcionan ninguna respuesta.					
<u>05-01</u>	Esperar a que el token de autenticación expire.					
<u>05-02</u>	Continuar usando el sistema sin ser redirigido para reautenticación.					
<u>05-03</u>	Ejecutar cualquier algoritmo, tratará de obtener el token, el cual ya ha vencido					
<u>BG-DJ-02</u>	El algoritmo de Dijkstra permite que se ingresen valores negativos como peso en los arcos del grafo, lo cual es incorrecto según la documentación estándar del algoritmo. Esto afecta la precisión de los resultados, ya que Dijkstra no garantiza que sea la distancia mínima o máxima cuando se emplean pesos negativos.	Al intentar ingresar un peso negativo en el arco, el sistema debería mostrar un mensaje de error indicando que los valores de peso deben ser positivos para asegurar la validez del cálculo de la distancia mínima o máxima.	El sistema permite el ingreso de pesos negativos en los arcos y procede a realizar el cálculo del algoritmo de Dijkstra sin alertar sobre el valor inválido, lo que produce resultados inexactos.	High	<u>BGDJ02.png</u>	@2024/10/28

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
<u>02-01</u>	Insertar un grafo en el dashboard dijkstra.					
<u>02-02</u>	Insertar arcos con pesos negativos.					
<u>02-03</u>	Ejecutar el algoritmo de Dijkstra en el grafo creado.					
<u>BG-GH-02</u>	Al intentar subir un archivo en la vista destinada para subir proyectos, no ocurre ninguna acción visible en la interfaz. Sin embargo, al inspeccionar la consola del navegador, se observa un error de servidor que impide completar la operación de carga del archivo.	Al hacer uso del botón para subir archivo el proyecto debería cargarse en la vista.	Al hacer clic en la interfaz no pasa nada y se genera un error en el backend.	High	<u>BGGH02.png</u>	@2024/12/14
<u>02-01</u>	El usuario accede a la vista destinada para subir proyectos.					
<u>02-02</u>	La vista se carga correctamente, mostrando el formulario y la opción para subir archivos.					

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
<u>02-03</u>	La vista se carga correctamente.					
<u>BG-JH-05</u>	Error al cargar archivos con algoritmos ya resueltos	Modificar un grafo a un nuevo estado válido todavía debería poder resolverse, aún después de recargarlo	Cuando se recarga un archivo que ya fue resuelto, modificarlo resulta en que ya no se puede resolver.	High	<u>JohnsonBug05.png</u>	@2024/10/15
<u>05-01</u>	Seleccionar un archivo de la lista y recargarlo					
<u>05-02</u>	Hacer cualquier modificación que resulte en un estado válido del grafo para ser resuleto por el algoritmo de Johnson					
<u>05-03</u>	Intentar resolver el algoritmo de ruta crítica					
<u>BG-NW-01</u>	El sistema no ajusta automáticamente la solución cuando la oferta y demanda están desbalanceadas, impidiendo resolver	El sistema debe detectar que la oferta y la demanda no están balanceadas y ajustar automáticamente la solución (por	El sistema no detecta que la oferta y la demanda no están balanceados, y no se ajusta la solución.	High	<u>Captura de pantalla 2024-10-28 a la(s) 20.22.33.png</u>	@2024/10/17

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	correctamente el problema de transporte.	ejemplo, añadiendo un nodo ficticio) para resolver el problema de transporte sin errores.				
<u>01-01</u>	Ingresar 100 unidades de oferta para los nodos de origen.					
<u>01-02</u>	Ingresar 80 unidades de demanda para los nodos de destino.					
<u>01-03</u>	Ejecutar el cálculo utilizando el método de la esquina noroeste.					
<u>BG-</u> <u>NW-02</u>	El sistema no ajusta automáticamente la solución cuando la oferta y demanda están desbalanceadas, impidiendo resolver correctamente el problema de transporte.	El sistema genera soluciones inconsistentes cuando la configuración de oferta y demanda no está balanceada, afectando la precisión de los resultados.	Si genera soluciones, aunque no siempre es válida.	High	<u>Captura de pantalla 2024-10-28 a la(s) 20.46.41.png</u> , <u>Captura de pantalla 2024-10-28 a la(s) 20.41.15.png</u> , <u>Captura de pantalla 2024-10-28 a la(s) 20.42.04.png</u>	@2024/10/17

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
<u>02-01</u>	Ingresar diferentes valores de oferta y demanda (balanceados y no balanceados).					
<u>02-02</u>	Ejecutar el cálculo del problema de transporte utilizando el módulo "Northwest".					
<u>BG-NW-03</u>	El sistema permite el envío de datos sin validar campos obligatorios ni valores numéricos, y no muestra mensajes de error en campos incorrectos o vacíos.	El sistema debe mostrar mensajes de error específicos indicando que los campos obligatorios están vacíos, que los valores en oferta y demanda deben ser numéricos, y que estos datos deben ser obligatorios.	El sistema no muestra mensajes de error específicos ni indica que los campos son obligatorios o que deben ser numéricos.	High	Captura de pantalla 2024-10-28 a la(s) 23.01.24.png , Captura de pantalla 2024-10-28 a la(s) 23.00.56.png , Captura de pantalla 2024-10-28 a la(s) 23.00.27.png	@2024/10/17
<u>03-01</u>	Intentar enviar el formulario con algunos campos vacíos.					
<u>03-02</u>	Ingresar datos no numéricos en los campos					

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	de oferta y demanda y enviar el formulario.					
<u>03-03</u>	Ingresar datos válidos en todos los campos y enviar el formulario, sin demanda y oferta.					
<u>BG-</u> <u>NW-05</u>	Después de ejecutar el algoritmo de la esquina noroeste, el editor se bloquea, impidiendo realizar modificaciones en el grafo. Solo vuelve a funcionar después de presionar Enter.	Después de ejecutar el algoritmo, el editor debe permitir al usuario modificar el grafo sin problemas.	El editor se bloquea y no permite realizar ninguna acción. Solo vuelve a funcionar al presionar Enter.	High	<u>Grabación de pantalla</u> <u>2024-10-28 a la(s)</u> <u>23.03.34.mov</u>	@2024/12/15
<u>05-01</u>	Solicitar la solución utilizando el método Northwest.					
<u>05-02</u>	Intentar volver al editor para hacer modificaciones en el grafo.					
<u>BG-ST-</u> <u>01</u>	Al ingresar una lista de valores numéricos negativos en el input del algoritmo de ordenamiento y ejecutar cualquiera de los	Se rompe los plots o diagramas en ciertos casos que se ingresa valores negativos	No se debería romper los plots en ningún conjunto de datos.	High	<u>IS-JAC-005.png</u>	@2024/10/28

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	algoritmos disponibles (Selection Sort, Insertion Sort, etc.), el gráfico resultante presenta problemas de visualización. Esto provoca que los elementos del gráfico se desplacen fuera de los límites de la pantalla, afectando la experiencia visual y de usuario.					
<u>01-01</u>	Accede a la aplicación en http://localhost:5173 o en el dominio desplegado.					
<u>01-02</u>	Dirígete a la sección de "Algoritmos de Ordenamiento" desde el menú de navegación.					
<u>01-03</u>	En el campo de entrada de números, ingresa una lista de valores numéricos negativos (por ejemplo: -1, -2, -5).					
<u>01-04</u>	Selecciona cualquiera de los algoritmos de					

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	ordenamiento disponibles.					
<u>BG-SU-01</u>	Al intentar registrar una cuenta dejando el campo de usuario vacío, el sistema muestra un mensaje indicando que la cuenta fue creada con éxito, pero a nivel de backend se registra un error de validación, y la cuenta no se crea correctamente.	El sistema le pide al usuario que llene todos los campos obligatorios	El sistema muestra que la cuenta se creo correctamente.	High	<u>Loom Evidencias</u>	@2024/12/12
<u>01-01</u>	El usuario accede al formulario de registro.					
<u>01-02</u>	El usuario deja el campo de usuario vacío y completa los demás campos correctamente.					
<u>01-03</u>	El usuario confirma el registro.					
<u>BG-SU-02</u>	Al intentar registrar una cuenta con una contraseña de menos de 6 caracteres, el sistema muestra un mensaje indicando que la cuenta fue creada con	El sistema le pide al usuario que su contraseña es muy corta y debería usar al menos 6 caracteres.	El sistema muestra que la cuenta se creo correctamente.	High	<u>Loom Evidencias</u>	@2024/12/12

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	éxito, pero a nivel de backend se registra un error de validación.					
<u>02-01</u>	El usuario accede al formulario de registro.					
<u>02-02</u>	El usuario ingresa una contraseña de menos de 6 caracteres y completa los demás campos.					
<u>02-03</u>	El usuario confirma el registro.					
<u>BG-SU-03</u>	En la vista de registro, al utilizar el botón para regresar a la página anterior, el sistema redirige al usuario a una ruta restringida que requiere inicio de sesión.	Debería mostrar la pantalla de Inicio nuevamente.	Se muestra pantallazo blanco debido a la falta de acceso, generando confusión para el usuario.	High	<u>Loom Evidencias</u>	@2024/12/12
<u>03-01</u>	El usuario accede a la vista de registro.					
<u>03-02</u>	El usuario hace clic en el botón "Página Anterior".					
<u>03-03</u>	El usuario intenta interactuar con la página					

Aa Codigo	≡ Descripción	➡ Resultado esperado	● Resultado actual	⚠ Severidad	📎 Evidencias	📅 Fecha deteccion
	tras el pantallazo.					

Anexo C - Bugs Resueltos

Aa Codigo	≡ Descripción	⚠ Severidad	📁 Evidencias	📁 Evidencia Corrección	📅 Fecha deteccion	📅 Fecha resolución	🕒 TMR
<u>BG-JH-01</u>	Cuando se crea una arista conectada a si misma, su peso no se mueve junto al elemento asociado	Low	<u>JohnsonBug01.png</u>	<u>BGJH-solved.png</u>	@2024/03/15	@2024/06/01	78
<u>BG-JH-02</u>	Error de grafos inválidos, vacíos y de conexiones incorrectas	Medium	<u>JohnsonBug02.png</u>	<u>BGJH-solved.png</u>	@2024/10/15	@2024/10/26	11
<u>BG-JH-03</u>	Error en grafos con un loop	Medium	<u>JohnsonBug03.png</u>	<u>BGJH-solved.png</u>	@2024/10/15	@2024/10/26	11

Anexo D - Reporte Análisis de Accesibilidad

Aa Codigo	≡ Descripción	* Principio afectado	⚠ Severidad	📎 Evidencias
<u>AX-JH-05</u>	La página debe contener un encabezado de nivel 1, tener un <i>main reference point</i> y estar asociada a un landmark (total 3 instancias) (está fuera de alcance del módulo)	- Principio: Perceptible - Criterio: 1.3.1 <i>Información y relaciones</i>	Low	<u>JohnsonEditor03.png</u>
<u>AX-JH-04</u>	En el editor de Johnson. Los elementos (botón se aceptar y resolver) deber tener un contraste de colores suficiente conforme a la WCAG 2 AA (1 instancia)	- Principio: Perceptible - Criterio: 1.4.3 <i>Contraste (mínimo)</i>	High	<u>JohnsonEditor02.png</u>
<u>AX-JH-03</u>	En el editor de Johnson. Los botones deben tener un texto discernible para lectores de pantalla. O bien como texto interno o aria-label (8 instancias)	- Principio: Operable - Criterio: 2.4.4 <i>Propósito de los enlaces (en contexto)</i> y 4.1.2 <i>Nombre, función, valo</i>	High	<u>JohnsonEditor01.png</u>
<u>AX-JH-02</u>	En el selector de archivos. La página debe contener un encabezado de nivel 1, tener un <i>main reference point</i> y estar asociada a un landmark (total 3 instancias) (está fuera de alcance del módulo)	- Principio: Perceptible - Criterio: 1.3.1 <i>Información y relaciones</i>	Low	<u>FileSelector02.png</u>

Aa Codigo	≡ Descripción	* Principio afectado	⚠ Severidad	📎 Evidencias
<u>AX-JH-01</u>	En la vista de selección de archivo, los elementos del formulario deben tener etiquetas (1 instancia)	- Principio: Perceptible - Criterio: 1.3.1 <i>Información y relaciones</i>	Medium	<u>FileSelector01.png</u>
<u>AX-GF-01</u>	Se identificó un problema de accesibilidad en el módulo de editor de grafos. El botón de retroceso (<) no tiene un texto interno visible para lectores de pantalla, lo que impide que los usuarios con discapacidades visuales comprendan su función.	- Principio: Operable - Criterio: 4.1.2 <i>Nombre, función, valor</i>	Low	<u>IS-JAC-003.png</u>
<u>AX-HM-01</u>	En la pantalla del listado de algoritmos, algunos botones presentan un problema de contraste insuficiente, lo cual dificulta su lectura y comprensión, especialmente para usuarios con discapacidades visuales.	- Principio: Perceptible - Criterio: 1.4.3 <i>Contraste (mínimo)</i>	Low	<u>IS-JAC-004.png</u>

Anexo E: Reporte de Defectos de Usabilidad

Cuadro 3

Plantilla de Reporte de Usabilidad

Items	Evaluation
1.- Visibilidad del estado del sistema	
¿Cada parte de la interfaz comienza con un título que describa el contenido de la pantalla?	Conforme
¿El diseño de íconos y su estética es consistente en todo el sistema?	Conforme
Cuando se selecciona un icono que está rodeado de otros iconos, ¿Se distingue claramente el ícono seleccionado?	No Conforme
Si se utilizan ventanas emergentes (pop-up) para mostrar mensajes de error, ¿Permiten esas ventanas que el usuario visualice el error en la interfaz cuando se despliegan?	No Conforme
¿Hay algún tipo de feedback para cada acción u operación?	No Conforme
Luego de que el usuario completa una acción o serie de acciones, ¿El "feedback" del sistema indica que el siguiente grupo de acciones puede completarse?	No Conforme
El sistema provee algún tipo de feedback visual en menús o cajas de diálogo que indiquen qué opciones pueden seleccionarse.	Conforme

El sistema provee algún tipo de feedback visual en menús o cajas de diálogo que indiquen en cuál de las posibles opciones se halla posicionado el cursor.	No Conforme
Si hay menús o caja de diálogo en donde pueden seleccionarse múltiples opciones, ¿El sistema provee algún tipo de "feedback" visual que indique cuáles son las opciones ya seleccionadas?	No Conforme
¿El sitio web entrega información corporativa de la organización?	Conforme
Si existen demoras mayores a 15 segundos en las respuestas del sistema, ¿El usuario es informado del progreso en la concreción de la respuesta?	No Conforme
¿Informa datos relevantes para quien no "navega"(Ej: Horas de atención)? ¿Y para hacer consultas web o no web (Ej: números de teléfono)?	No Conforme
¿Los tiempos de respuesta son apropiados para cada tarea?	Conforme
Tiempo de escritura, movimiento del cursor o selección con el ratón: entre 0,5 y 1,5 milisegundos	Conforme
Tareas más comunes: 2 a 4 segundos	Conforme
Tareas complejas: 8 a 12 segundos	Conforme
No son necesarios altos niveles de concentración y no es requerido retener información: 2 a 15 segundos	No Conforme
La terminología usada en los menús, ¿Es consistente con el dominio de conocimiento del usuario en relación a la tarea a realizar?	Conforme

¿El usuario conoce su ruta de ubicación?	Conforme
2.- Relación entre el sistema y el mundo real	
¿Los íconos son concretos y familiares para el usuario?	Conforme
¿Los colores seleccionados corresponden a los valores esperados?	No Conforme
Cuando se ingresan datos en la pantalla, ¿La terminología utilizada para describir la tarea es familiar para los usuarios?	Conforme
Cuando la pantalla incluye preguntas, ¿El lenguaje de esas preguntas es claro y conciso?	N/A
Las combinaciones de secuencias de letras o palabras extrañas o poco frecuentes, ¿Se evitan siempre que sea posible?	No Conforme
El sistema ingresa/elimina de manera automática los signos de pesos o dólar y decimal cuando se insertan valores monetarios.	N/A
¿Se utilizan nombres unívocos y descriptivos en todo momento?	No Conforme
¿Se hace uso de los rastreadores de progreso?	No Conforme
Los H1 están optimizados para SEO	No Conforme
3.- Control y libertad por parte del usuario	
En sistemas que permitan el uso de ventanas superpuestas ¿Es fácil reacomodar reubicar esas ventanas en la pantalla?	Conforme

En sistemas que permitan el uso de ventanas superpuestas ¿Es fácil para los usuarios cambiar de una ventana a otra?	Conforme
Cuándo una tarea efectuada por el usuario se completa ¿el sistema espera alguna señal del usuario antes de procesar la tarea?	Conforme
¿Se pregunta al usuario que confirme acciones que tendrán consecuencias drásticas, negativas o destructivas?	No Conforme
¿Existe una función para "deshacer." a1 nivel de cada acción simple, cada entrada de datos y cada grupo de acciones completadas?	No Conforme
¿Los usuarios pueden cancelar acciones en progreso?	No Conforme
¿Los usuarios pueden reducir el tiempo de entrada de datos copiando y modificando datos existentes?	Conforme
Los menús son anchos (muchos ítems), antes que profundos (muchos niveles)	No Conforme
Si el sistema posee menús de niveles múltiples ¿Existe algún mecanismo que permita a los usuarios regresar al menú previo?	Conforme
Los usuarios pueden moverse hacia delante o hacia atrás entre las opciones de campos o cajas de dialogo.	Conforme
Si el sistema utiliza una interfaz de preguntas y respuestas ¿Pueden los usuarios regresar a la pregunta anterior o saltar hacia delante una pregunta?	N/A

¿Los usuarios pueden revertir sus acciones de manera sencilla?	Conforme
Si el sistema permite a los usuarios revertir sus acciones , ¿Existe un mecanismo que permita "deshacer" varias acciones de manera simultánea?	No Conforme
4.- Consistencia y estándares	
El abuso de letras en mayúscula en la pantalla se ha evitado	No Conforme
No hay más de 12/20 tipos de íconos	Conforme
Existe algún elemento visual que identifique la ventana activa	Conforme
Cada ventana posee un título	Conforme
¿Es posible utilizar las barras de desplazamiento horizontal y vertical en cada ventana?	Conforme
Si una opción de un menú es la de "salir" ¿Esta opción aparece como ultimo ítem en el menú?	No Conforme
¿Los títulos de los menús están centrados o justificados a la izquierda?	No Conforme
Fuentes: hasta tres tipos como máximo	No Conforme
Hasta cuatro colores (usados ocasionalmente)	No Conforme
Sonido: tonos suaves para dispositivos de retroalimentación ocasional y bruscos para condiciones críticas.	N/A
¿Se provee una leyenda si los códigos de color son numeros o difíciles de interpretar?	No Conforme

Se evitan los pares de colores espectralmente extremos y altamente cromáticos	No Conforme
Los azules saturados no se utilizan para texto u otro elemento pequeño.	No Conforme
La información más importante esta above the fold (la parte del sitio que los usuarios ven primero)	No Conforme
¿La estructura de la entrada de datos es consistente entre las diferentes pantallas?	No Conforme
5.- Prevención de errores	
¿Las entradas de datos no son sensibles a mayúsculas siempre que sea posible?	No Conforme
Las pantallas para entrada de datos y cajas de diálogo indican el número de espacios en caracteres que estan disponibles para un campo	No Conforme
Los campos en las pantallas de entrada de datos y las cajas de diálogo ¿contienen valores por defecto cuando corresponden?	No Conforme
6.- Reconocer antes que recordar	
¿Las áreas de texto tienen "espacios de respiración" que las rodeen?	No Conforme
¿Se ha utilizado el mismo color para agrupar elementos relacionados?	No Conforme
¿Existe buen contraste de brillo y de color entre los colores usados para imágenes y fondos?	No Conforme

Los colores suaves, brillantes y saturados se han utilizado para enfatizar datos, mientras que los colores oscuros, opacos y no saturados, han sido usados para des-enfatizar datos?	No Conforme
¿Los ítems inactivos en un menú aparecen en gris o están omitidos?	No Conforme
7.- Flexibilidad y eficiencia en el uso	
Los usuarios pueden reducir el tiempo de entrada de datos si se les permite copiar y pegar datos existentes.	Conforme
Si las listas de menú son cortas (siete ítem o menos) ¿Pueden los usuarios seleccionar un ítem moviendo el cursor?	Conforme
8.- Diseño estético y minimalista	
Los íconos son visuamente distinguibles de acuerdo a su significado conceptual	Conforme
¿Cada ícono esta resaltado con respecto a su fondo?	Conforme
Cada pantalla de entrada de datos incluye un título simple, corto, claro y suficientemente distintivo.	No Conforme
Los títulos de los menús son breves pero lo suficientemente largos como para comunicar su contenido.	No Conforme
9.- Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	
¿Los sonidos son utilizados para señalar errores?	N/A

Si se usan mensajes de error con humor ¿Son apropiados y respetuosos para la comunidad de usuarios?	No Conforme
¿Los mensajes de error son gramaticalmente correctos?	Conforme
¿Los mensajes de error evitan el uso de signos de admiración?	No Conforme
Los mensajes de error evitan el uso de palabras violentas u hostiles	No Conforme
Si se detecta un error en un campo de entrada de datos ¿El sistema posiciona el cursor en ese campo o lo resalta de alguna manera?	No Conforme
¿Los mensajes de error sugieren la causa del problema que lo has ocasionado?	No Conforme
¿Los mensajes de error indican que acción debe realizar el usuario para corregir el error correspondiente?	No Conforme
10.- Ayuda y documentación	
¿Las instrucciones en línea se distinguen visualmente?	No Conforme
Si las opciones de los menús son ambiguas ¿el sistema provee información aclaratoria adicional cuando un ítem es seleccionado?	No Conforme
¿La función de ayuda del menú es visible? (Por ejemplo una tecla etiquetada AYUDA o un menú especial)	No Conforme
Navegación: la información es fácil de encontrar	Conforme
¿La información es exacta, completa y comprensible? ¿La información es relevante?	No Conforme

Tras haber accedido a la ayuda ¿Pueden los usuarios continuar con su trabajo desde donde ha sido interrumpido?	Conforme
¿Es fácil acceder y regresar del sistema de ayuda?	Conforme