



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	2
Evaluación de riesgos	2
Riesgos del Producto	2
Riesgos de Proyecto	5
Evaluación de defectos	1
Accesibilidad y usabilidad	2
Resultados Obtenidos en Pruebas de Usabilidad	2
Conclusiones	3
Lecciones Aprendidas y Recomendaciones	4
Anexos	6
Anexo A: Reporte de Defectos de Usabilidad	6

Índice de figuras

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

Índice de tablas

1.	Cronograma Final del Proyecto	4
2.	Tiempo Medio de Reparación de Bugs	9
3.	Plantilla de Reporte de Usabilidad	6

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).

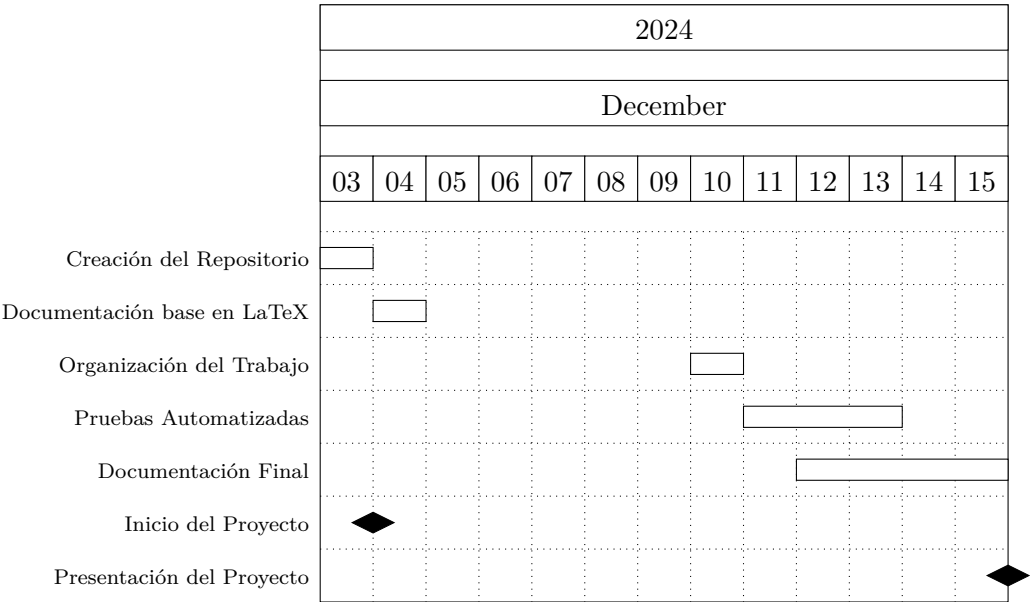
Cuadro 1*Cronograma Final del Proyecto*

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.

Este porcentaje muestra un alto grado de estabilidad en el sistema, aunque los casos fallidos indican áreas que requieren corrección para garantizar un funcionamiento completo.
- **Defectos identificados:** Durante las pruebas, se encontraron un total de **30 defectos**. De estos, se corrigieron **3**, representando un **10 % de bugs resueltos**. *El bajo porcentaje de resolución de defectos sugiere que aún se necesita priorizar la corrección de errores para mejorar la calidad y estabilidad del sistema.*
- **Tiempo total de ejecución:** Las pruebas requirieron un total de **15 horas-persona**. *Este tiempo refleja un esfuerzo considerable en la ejecución de pruebas y demuestra la dedicación del equipo para validar el funcionamiento del sistema dentro de los plazos establecidos.*
- **Cobertura de pruebas:**

 - **Historias de Usuario cubiertas:** 26 *La cobertura del 79 % de las historias de usuario indica un alcance robusto de las pruebas funcionales en las funcionalidades clave del sistema.*
 - **Historias de Usuario no cubiertas:** 7 *Las historias de usuario no cubiertas representan áreas pendientes que deben ser validadas en futuras iteraciones para garantizar la completitud del producto.*
- **Automatización de Test Cases:**

- **Usando Postman:** 4 casos exitosos, 0 fallidos. *Los resultados de Postman reflejan que las pruebas automatizadas para las APIs se ejecutaron con éxito y sin errores, lo que indica estabilidad en los servicios backend.*
- **Usando Playwright:** 6 casos exitosos, 2 fallidos. *El desempeño en Playwright sugiere que la automatización de pruebas de interfaz es efectiva en su mayoría, aunque los casos fallidos resaltan problemas específicos que deben abordarse.*

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:

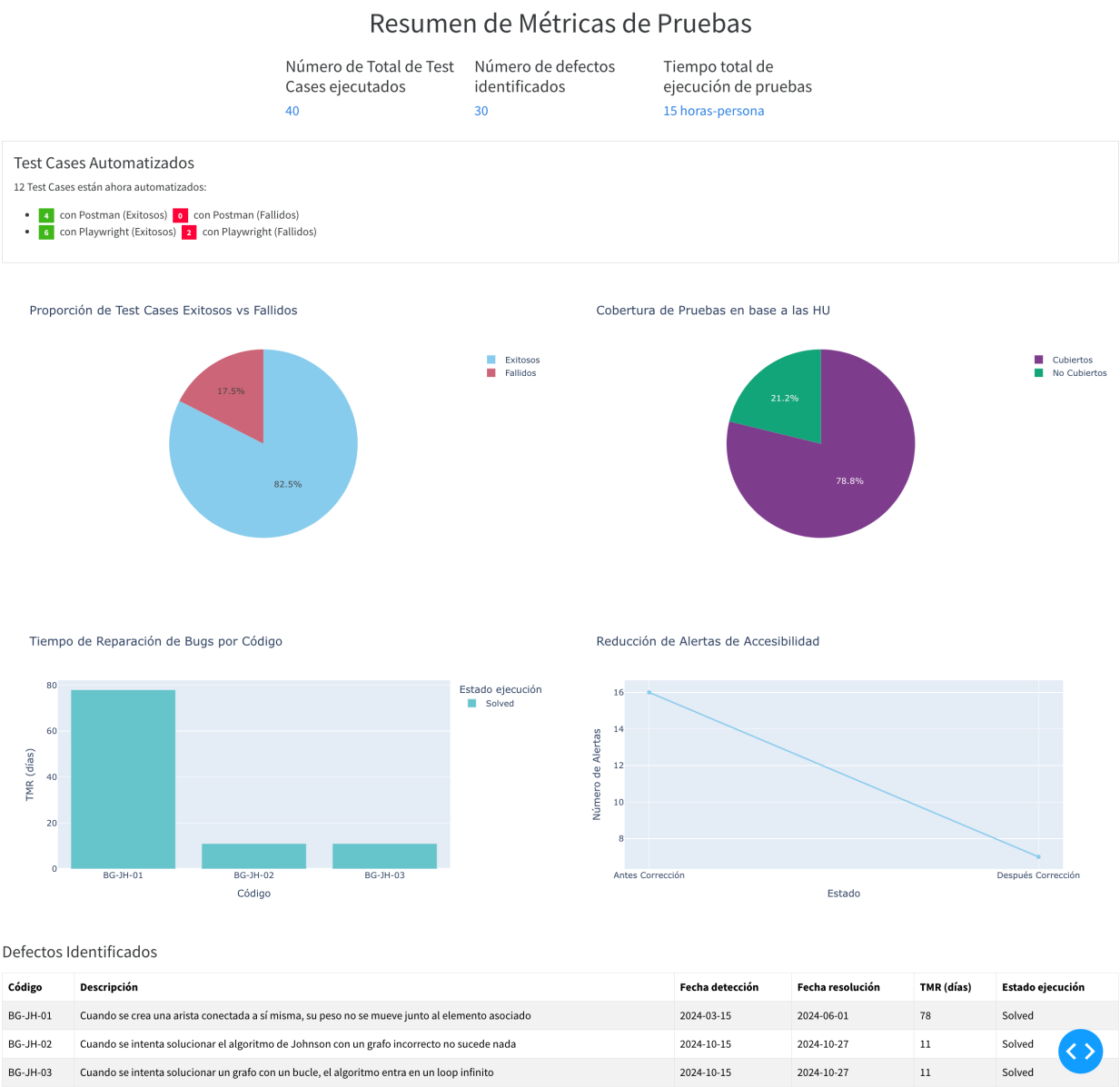


Figura 2

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 (**ver anexo ??**). 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 Anexos)

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.

Anexos

Anexo A: 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 ha 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