

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	<ul style="list-style-type: none"> - 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-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-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 critica 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: "edgId", "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.			El 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</u> , <u>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	- 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>	Ingresa 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>	Ingresa 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.		<p>contenedor.</p> <ul style="list-style-type: none"> -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	<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 - 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
<u>11-04</u>	Repetir los pasos 1 a 3 hasta tener un grafo similar a la referencia			(ver evidencia anexa)		Passed
<u>11-05</u>	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
<u>11-06</u>	Ingresar un valor numerico y guardar cambios			El modal se cierra y la arista ahora tiene ese valor		Passed
<u>11-07</u>	Repetir los pasos <u>11-01</u> a <u>11-06</u> hasta construir un grafo direccionado con un nodo inicial y un nodo final sin generar bucles			Ver evidencia		Passed
<u>TC-GF-03</u>	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. 		<u>TC-GF-03.png</u>	Passed
<u>03-01</u>	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.		
<u>03-07</u>	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.		
<u>03-08</u>	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.		
<u>03-09</u>	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.		
<u>03-10</u>	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	<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 		<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>	Ingresa 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.		<u>DijkstraCorrectTCDJ01.jpg</u>	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	<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>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.		