

Formato de escenarios y casos de uso

Grupo JCD

Configuración de los Escenarios

Nombre	Clase	Escenario
setupStage1	PersonaTest	El sistema inicia con un conjunto vacío de personas.
setupStage2	PersonaTest	El sistema inicia con un conjunto de personas que contiene 2 registros: P001 (Juan, 3101234567), P002 (María, 3129876543).
setupStage3	IncidenteTest	Existen 2 incidentes cargados con fechas diferentes: I001 (robo, 01-03-2025 12:00), I002 (accidente, 01-03-2025 15:00).
setupStage4	RutaTest	Existen 2 rutas registradas: R001 (distancia=10km, tiempo=15min), R002 (distancia=5km, tiempo=10min).
setupStage5	ImportExportTest	Hay un archivo JSON válido llamado "datos_personas.json" que contiene 3 personas, y un archivo "datos_rutas.json" con 2 rutas válidas.
setupStage6	ReporteTest	El sistema cuenta con listas ya ordenadas de rutas e incidentes, y un conjunto de 3 personas para generar un reporte consolidado y estadísticas.

Diseño de Casos de Prueba

RF-01: Gestión de Archivos JSON

Objetivo de la Prueba: Validar la correcta importación y exportación de datos en formato JSON.				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
ImportExportTest	importarPersonas(String filePath)	setupStage5	filePath = "datos_personas.json"	El sistema carga 3 personas al conjunto; no se producen errores.
ImportExportTest	importarRutas(String filePath)	setupStage5	filePath = "datos_rutas.json"	El sistema carga 2 rutas en la lista; no se producen errores.

RF-02: Manejo de Excepciones

Objetivo de la Prueba: Verificar el manejo adecuado de excepciones durante la importación de archivos.				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
ImportExportTest	importarPersonas(String filePath)	setupStage5	filePath = "no_existe.json"	Se lanza FileNotFoundException (o excepción personalizada) y no se altera el conjunto.
ImportExportTest	importarPersonas(String filePath)	setupStage5	filePath = "datos_corruptos.json"	Se lanza excepción de formato (JSONException); no se agregan datos al conjunto.

RF-03: Almacenamiento y Manipulación mediante Listas Enlazadas

Objetivo de la Prueba: Confirmar que las operaciones de inserción y eliminación funcionen en listas enlazadas.				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
ListaEnlazadaTest	insert(Object data)	setupStage4	data = Ruta(ID="R003", dist=8, t=12)	La lista de rutas incrementa su tamaño en 1 y se puede recuperar la ruta "R003".
ListaEnlazadaTest	remove(Object data)	setupStage4	data = Ruta(ID="R002")	La lista decreenta su tamaño en 1 y la ruta "R002" ya no se encuentra en la estructura.

RF-04: Registro y Gestión de Rutas

Objetivo de la Prueba: Verificar el registro de rutas y su ordenamiento.				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
RutaTest	registrarRuta(Ruta ruta)	setupStage4	ruta = (ID="R003", dist=12, tiempo=20, inicio="A", fin="B")	Se añade "R003" a la lista de rutas; tamaño pasa de 2 a 3.
RutaTest	ordenarRutasPorDistancia()	setupStage4	N/A	La lista queda ordenada: primero "R002" (5km), luego "R001" (10km).

...

RF-05: Registro y Gestión de Incidentes

Objetivo de la Prueba: Verificar el registro de incidentes y su ordenamiento por fecha/hora.				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
IncidenteTest	registrarIncidente(Incidente inc)	setupStage3	inc = (ID="I003", tipo="Robo", fecha="01-03-2025 09:00", estado="Pend.")	Se añade "I003"; la lista de incidentes pasa de 2 a 3.
IncidenteTest	ordenarIncidentesPorFecha()	setupStage3	N/A	Se ordena la lista, primero el incidente más reciente (I002), luego I001 e I003.

...

RF-06: Búsqueda de Elementos y Determinación de la Mejor Ruta

Objetivo de la Prueba: Verificar la búsqueda de incidentes por ID y la selección de la ruta óptima.				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
BusquedaTest	buscarIncidente(String id)	setupStage3	id="I001"	Se retorna el incidente con ID="I001".
RutaTest	determinarMejorRuta()	setupStage4	N/A (usa la lista de rutas)	Se retorna la ruta con menor distancia y menor tiempo (en este ejemplo, R002).

...

RF-07: Registro y Validación de Personas

Objetivo de la Prueba: Verificar el registro de personas y la validación de unicidad del ID.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
PersonaTest	registrarPersona(Persona p)	setupStage1	p = (ID="P003", nombre="Carlos", contacto="3101234567")	Se añade "P003" al conjunto vacío; tamaño pasa de 0 a 1.
PersonaTest	registrarPersona(Persona p)	setupStage2	p = (ID="P002", nombre="María", contacto="3129876543")	Se rechaza o lanza excepción por ID duplicado; tamaño se mantiene en 2.

...

RF-08: Actualización y Eliminación de Personas

Objetivo de la Prueba: Verificar la actualización y eliminación de personas en el conjunto.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
PersonaTest	actualizarPersona(String id, Persona nueva)	setupStage2	id="P002", nueva: nombre="María López", contacto="3121112222"	Se actualizan los datos de "P002"; al buscarla, se obtienen los datos nuevos.
PersonaTest	eliminarPersona(String id)	setupStage2	id="P001"	El conjunto pasa de 2 a 1; "P001" ya no aparece en la búsqueda.

...

RF-09: Operaciones Complementarias sobre el Conjunto de Personas

Objetivo de la Prueba: Validar la impresión, conteo, conversión a arreglo y limpieza del conjunto.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
OperacionesConjuntoTest	imprimirConjunto()	setupStage2	N/A	Se imprime en consola la lista con "P001" y "P002".
OperacionesConjuntoTest	convertirAArreglo()	setupStage2	N/A	Se obtiene un arreglo con 2 posiciones; luego se puede invocar limpiarConjunto() para dejarlo vacío.

...

RF-10: Ordenamiento y Filtrado de Personas

Objetivo de la Prueba: Verificar el ordenamiento por nombre y el filtrado por edad (u otro criterio).				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
PersonaTest	ordenarPorNombre()	setupStage2	N/A	Se ordena alfabéticamente: primero "Juan" (P001), luego "María" (P002).
PersonaTest	filtrarPorEdad(int edad)	setupStage2	edad = 30	Retorna las personas con edad ≥ 30 (si las hay); si no, lista vacía.

...

RF-11: Importación y Exportación del Conjunto de Personas

Objetivo de la Prueba: Verificar la importación y exportación del conjunto de personas.				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
ImportExportTest	importarPersonas(String filePath)	setupStage5	filePath = "datos_personas.json"	Se cargan 2 personas en el conjunto; no se producen errores.
ImportExportTest	exportarPersonas(String filePath)	setupStage5	filePath = "export_personas.json"	Se genera un archivo JSON con las personas actuales; no se producen errores.

...

RF-12: Generación de Reportes y Estadísticas Integrales

Objetivo de la Prueba: Verificar la creación de reportes consolidados y el cálculo de estadísticas.				
Clase	Método	Escenario	Valores de Entrada	Resultado esperado
ReporteTest	generarReporte()	setupStage6	Listas de rutas, incidentes y 2 personas ya cargadas	Se muestra (o se genera archivo) con la info ordenada de rutas, incidentes y personas.
ReporteTest	calcularEstadisticas()	setupStage6	Datos de rutas (distancias), incidentes (número) y personas (edades)	Retorna indicadores (por ejemplo, total incidentes, promedio de distancia, promedio de edad, etc.).

...