Formato de escenarios y casos de uso

Configuración de los Escenarios

Nombre	Clase	Escenario

setupStage1	PassengerTest	Passenger != null
		addPassenger [
		{
		“passengerID”: “132532”,
		“name”: “José”,
		“assignedRoute”: “Ruta2”,
		“contact”: “3154322674”;
		}
		]

setupStage2	JSONTest	A JSON file with passenger information is written and read

setupStage3	DriverTest	Driver != null
		addDriver [
		{
		“driverID”: “246731”,
		“name”: “Andres”,
		“assignedVehicle”: “car / motorcycle…”,
		“state”: “available / on route”;
		}
		]

setupStage4	DriverTest	Driver != null
		
		searchDriver(name) [
		{
		nombre != null
		
		“nameDriver”: “Andres”,
		“Driver found”;
		}
		]

setupStage5	JSONTest	A JSON file with driver information is written and read.

setupStage6	RouteTest	Route != null
		addRoute [
		{
		“routeID”: “823152”,
		“distance”: “33km”,
		“stimatedTime: “2 horas”,
		“startPoint”: “Punto 12”,
		“endPoint”: “Punto 17”;
		}
		]

setupStage7	RouteTest	Route != null
		
		orderOfRoutesByDistance(distance) [
		{
		“route1”: “5 km”,
		“route2”: “12 km”,
		“route3”: “20 km”;
		}
		]

setupStage8	RouteTest	Route != null
		
		orderByEstimatedTime(time) [
		{
		“routeTime1”: “30 min”,
		“routeTime2”: “58 min”,
		“routeTime3”: “1 hour”;
		}
		]

setupStage9	RouteTest	Route != null
		
		bestRoute(distance, time) [
		{
		“bestRoute”: “route1”;
		}
		]

setupStage10	JSONTest	A JSON file with route information is written and read.

setupStage11	VehicleTest	Vehicle != null addVehicle
		[
		{
		“vehicleID”: “165243”,
		“vehicleColor”: “red / blue…”,
		“Type”: “car / motorcycle…”;
		}
		]

setupStage12	IncidentTest	Incident != null
		addIncident [
		{
		“incidentID”: “652413”,
		“Type”: “theft / accident / fire…”,
		“location”: “coordinates / address”,
		“date”: “02/08/2025”,
		“hour”: “14:33 pm”,
		“description”: “Incident on 32nd Street due to theft of a red car…”,
		“state”: “pending / in process / resolved”;
		}
		]

setupStage13	IncidentTest	Incident != null
		
		orderByDateAndTime(date, time) [
		{
		“incident1”: “04/04/2025, 14:33 pm”,
		“incident2”: “24/10/2025, 10:13 am”,
		“incident3”: “11/06/2025, 06:20 am”;
		}
		]

setupStage14	IncidentTest	Incident != null
		
		searchIndicent(ID) [
		{
		incidentID != null
		“incidentID”: “652413”
		“Incident found”;
		}
		]

setupStage15	JSONTest	A JSON file with incident information is written and read.

setupStage16	ReportTest	Report != null
		
		reportGeneration(route, incident) [
		{
		“Route”: “route2”,
		“Incident”: “accident”;
		
		“Route”: “route4”,
		“Incident”: “theft”;
		
		…
		}
		]

setupStage17	MonitoringTest	Monitoring != null
		
		emergencyVehicleMonitoring() [
		{
		“Vehicle”: “ambulance”,
		“Location”: “5th avenue north”;
		
		“Vehicle”: “police patrol”,
		“Location”: “cañasgordas avenue”;
		
		…
		}
		]

setupStage18	TrafficTest	Traffic != null
		
		stateOfTrafficAndTransport() [
		{
		“trafficStatus”: “congested”,
		“transportStatus”: “belated”;
		
		…
		}
		]

setupStage19	NotificationTest	Notification != null
		
		notificationToCitizens() [
		{
		“Incident”: “accidente”,
		“alternateRoutes”: “route2”,
		“recommendations”: “avoid the area”;
		
		…
		}
		]

Universidad Icesi
Departamento de Computación y Sistemas Inteligentes
Algoritmos y Programación II

Diseño de Casos de Prueba

Objetivo de la Prueba: Verify that a passenger can be registered correctly, both positively and negatively.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado

Passenger	addPassenger	setupStage1	{
			“passengerID”: “132532”,
			“name”: “José”,
			“assignedRoute”: “Ruta2”,
			“contact”: “3154322674”;
			}
				"Passenger registered successfully."

Passenger	addPassenger	setupStage1	{
			"passengerID": "",
			"name": "José",
			"assignedRoute": "Ruta2",
			"contact": "3154322674"
			}
				"Error: Passenger ID cannot be empty."

JSONTest	writeReadJSON	setupStage2	JSON with passengers	“Data correctly read from JSON”
JSONTest	writeReadJSON	setupStage2	JSON with passengers	“Data incorrectly read from JSON”

Objetivo de la Prueba: Check that a driver can be registered correctly and search for him/her name, both positively and negatively.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado

Driver	addDriver	setupStage3	{
			“driverID”: “246731”,
			“name”: “Andres”,
			“assignedVehicle”: “car / motorcycle…”,
			“state”: “available / on route”;
			}
				“Driver registered successfully.”

Driver	addDriver	setupStage3	{
			“driverID”: “246731”,
			“name”: “”,
			“assignedVehicle”: “car / motorcycle…”,
			“state”: “available / on route”;
			}
				"Error: Driver name cannot be empty."

Driver	searchDriver	setupStage4	“nameDriver”: “Andres”	“Driver found”
Driver	searchDriver	setupStage4	“nameDriver”: “Jorge”	“Driver not found.”

JSONTest	writeReadJSON	setupStage5	JSON with drivers	“Data correctly read from JSON”
JSONTest	writeReadJSON	setupStage5	JSON with drivers	“Data incorrectly read from JSON”

Objetivo de la Prueba: Verify that a route can be registered correctly. sort routes by distance, estimated time and determine the best route according to defined criterio, both positively and negatively.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Route	addRoute	setupStage6	{ “routeID”: “823152”, “distance”: “33km”, “stimatedTime: “2 horas”, “startPoint”: “Punto 12”, “endPoint”: “Punto 17”;	“Route registered correctly.”
			}	

Route	addRoute	setupStage6	{
			“routeID”: “823152”,
			“distance”: “33km”,
			“stimatedTime: “”,
			“startPoint”: “Punto 12”,
			“endPoint”: “Punto 17”;
			}
				"Error: Estimated time cannot be empty."

Route	orderOfRoutesBy Distance	setupStage7	{
			“route1”: “5 km”,
			“route2”: “12 km”,
			“route3”: “20 km”;
			}
				“List of routes sorted by distance.”

Route	orderOfRoutesBy Distance	setupStage7	{
			“route1”: “5 km”,
			“route2”: “-12 km”,
			“route3”: “20 km”;
			}
				"Error: Distance cannot be negative."

Route	orderByEstimated Time	setupStage8	{
			“routeTime1”: “30 min”,
			“routeTime2”: “58 min”,
			“routeTime3”: “1 hour”;
			}
				“List of routes ordered by time.”

Route	orderByEstimated Time	setupStage8	{
			“routeTime1”: “30 min”,
			“routeTime2”: “RE min”,
			“routeTime3”: “1 hour”;
			}
				"Error: Invalid time format."

Route	bestRoute	setupStage9	{
			“bestRoute”: “route1”;
			}
				“Optimal route according to selected criterio.”

Route	bestRoute	setupStage9	{
			“bestRoute”: “”;
			}
				"Error: No valid routes available."

JSONTest	writeReadJSON	setupStage10	JSON with routes	“Data correctly read from JSON”
JSONTest	writeReadJSON	setupStage10	JSON with routes	“Data incorrectly read from JSON”

Objetivo de la Prueba: Verify that a vehicle can be registered correctly, both positively and negatively.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado

Vehicle	addVehicle	setupStage11	{
			“vehicleID”: “165243”,
			“vehicleColor”: “red / blue…”,
			“Type”: “car / motorcycle…”;
			}
				“Vehicle registered correctly.”

Vehicle	addVehicle	setupStage11	{ “vehicleID”: “165243”, “vehicleColor”: “red / blue…”, “Type”: “airplane”;	"Error: Invalid vehicle type. allowed types: car/motorcycle."
			}	

Objetivo de la Prueba: Verify that an incident can be recorded correctly. sort incident by date/time and search for an incident by ID, both positively and negatively.

Clase	Método	Escenario	Valores de Entrada	Resultado esperado

Incident	addIncident	setupStage12	{
			“incidentID”: “652413”,
			“Type”: “theft / accident / fire…”,
			“location”: “coordinates / address”,
			“date”: “02/08/2025”,
			“hour”: “14:33 pm”,
			“description”: “Incident on 32nd Street due to theft of a red car…”,
			“state”: “pending / in process / resolved”;
			}
				“Incident registered correctly.”

Incident	addIncident	setupStage12	{
			“incidentID”: “652413”,
			“Type”: “theft / accident / fire…”,
			“location”: “coordinates / address”,
			“date”: “DOS/AGOSTO/2025”,
			“hour”: “14:33 pm”,
			“description”: “Incident on 32nd Street due to theft of a red car…”,
			“state”: “pending / in process / resolved”;
			}
				"Error: Date format is incorrect."

Incident	orderByDate AndTime	setupStage13	{
			“incident1”: “04/04/2025, 14:33 pm”,
			“incident2”: “24/10/2025, 10:13 am”,
			“incident3”: “11/06/2025, 06:20 am”;
			}
				“List of incidents sorted by date and hour.”

Incident	orderByDate AndTime	setupStage13	{
			“incident1”: “04/04/2025, 14:33 pm”,
			“incident2”: “SA/10/XXXX, 10:13 am”,
			“incident3”: “11/XX/2025, 06:20 am”;
			}
				"Error: Invalid date format."

Incident	searchIncident	setupStage14	“incidentID”: “652413”	“Inicident found.”

Incident	searchIncident	setupStage14	"incidentID": "999999"	"Error: Incident not found."

JSONTest	writeReadJSON	setupStage15	JSON with incidents	“Data correctly read from JSON”
JSONTest	writeReadJSON	setupStage15	JSON with incidents	“Data incorrectly read from JSON”

Objetivo de la Prueba: Verify the generation of route and incident reports, both positively and negatively

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
Report	reportGenerator	setupStage16	{ “Route”: “route2”, “Incident”: “accident”;	“Reports generated correctly.”
			“Route”: “route4”, “Incident”: “theft”; … }]	

Report	reportGenerator	setupStage16	{
			“Route”: “routeH”,
			“Incident”: “accident”;
			
			“Route”: “routeE”,
			“Incident”: “theft”;
			
			…
			}
			]
				"Error: Route does not exist."

Objetivo de la Prueba: Monitor the location of emergency vehicles, both positively and negatively

Clase	Método	Escenario	Valores de Entrada	Resultado esperado

Monitoring	emergencyVehicleMonitoring	setupStage17	{
			“Vehicle”: “ambulance”,
			“Location”: “5th avenue north”;
			
			“Vehicle”: “police patrol”,
			“Location”: “cañasgordas avenue”;
			
			…
			}
				“Vehicle location updated in real time.”

Monitoring	emergencyVehicleMonitoring	setupStage17	{
			“Vehicle”: “ambulance”,
			“Location”: “5th avenue north”;
			
			“Vehicle”: “police patrol”,
			“Location”: “”;
			
			…
			}
				"Error: Location cannot be empty."

Objetivo de la Prueba: Get real-time traffic and transportation status, both positively and negatively

Clase	Método	Escenario	Valores de Entrada	Resultado esperado

Traffic	stateOfTrafficAndTransport	setupStage18	{
			“trafficStatus”: “congested”,
			“transportStatus”: “belated”;
			
			…
			}
				“Real-time information processed.”

Traffic	stateOfTrafficAndTransport	setupStage18	{
			“trafficStatus”: “stopped”,
			“transportStatus”: “uknown”;
			
			…
			}
				"Error: Invalid transport status."

Objetivo de la Prueba: Send notifications of incidents, alternate routes and recommendations, both positively and negatively

Clase	Método	Escenario	Valores de Entrada	Resultado esperado

Notification	notificationToCitizens	setupStage19	{
			“Incident”: “accidente”,
			“alternateRoutes”: “route2”,
			“recommendations”: “avoid the area”;
			
			…
			}
				“Notification sent successfully.”

Notification	notificationToCitizens	setupStage19	{
			“Incident”: “accidente”,
			“alternateRoutes”: “”,
			“recommendations”: “avoid the area”;
			
			…
			}
				"Error: At least one alternate route must be provided."