

[Enlace documento original - Acercar y ver los diagramas](#)

Presentación pruebas unitarias JUnit


A continuación se presenta el desarrollo a realizar de las pruebas unitarias de los métodos del sistema para el nuevo aeropuerto internacional de Madrid, se presentan los escenarios y los casos. Con el propósito de facilitar el entendimiento de la aplicación se presentan las pruebas a realizar en cada uno de los contextos marcados por el tipo de usuario que usa la aplicación.

Administrador General Aeropuerto - Casos

Crear nuevo usuario administrador del aeropuerto

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en un solo caso, añadir correctamente el objeto la relación está null.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario1	Airport	 <pre>graph LR Airport[":Airport"] -- mainUser --> null["null"]</pre>

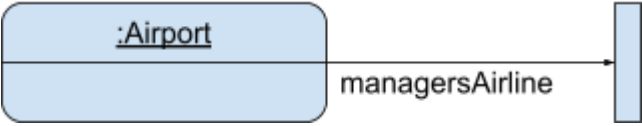
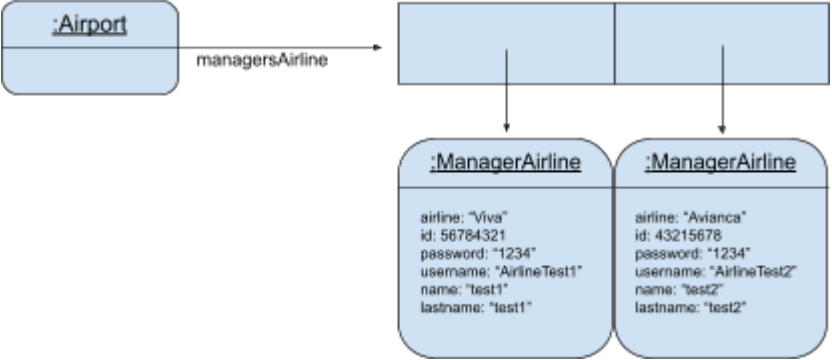
Diseño de Casos de Prueba

Objetivo de la Prueba: Registrar un nuevo usuario administrador general.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	createMainUser	setupScenario1	username:"general IMG" name: "test" lastname: "testL" password:"1234"	User created successfully. Returns the name of the main user and the ID

Crear nuevo usuario administrador de aerolínea

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en 2 casos, añadir correctamente el objeto al arreglo está vacío y cuando hay otros objetos en el.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario2	Airport	 <pre> sequenceDiagram participant Airport as :Airport participant Array Airport->>Array: managersAirline </pre>
setupScenario3	Airport	 <pre> sequenceDiagram participant Airport as :Airport participant Array participant ManagerAirline1 as :ManagerAirline participant ManagerAirline2 as :ManagerAirline Airport->>Array: managersAirline Array-->>ManagerAirline1: Array-->>ManagerAirline2: ManagerAirline1->>Note1: airline: 'Viva', id: 56784321, password: '1234', username: 'AirlineTest1', name: 'test1', lastname: 'test1' ManagerAirline2->>Note2: airline: 'Avianca', id: 43215678, password: '1234', username: 'AirlineTest2', name: 'test2', lastname: 'test2' </pre>

Diseño de Casos de Prueba

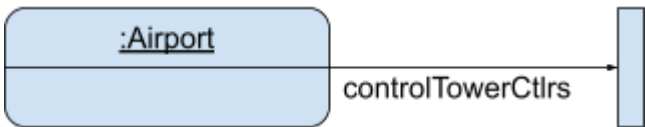
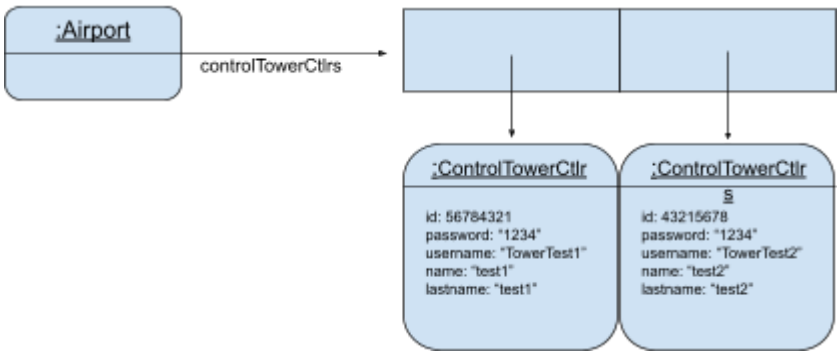
Objetivo de la Prueba: Registrar un nuevo usuario administrador de aerolínea				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	createAirlineManager	setupScenario2	airline: "Copa" id: 12345678 password: "1234" username: "AirlineTest" name: "test" lastname: "testL"	User created successfully. Returns Copa, 12345678, 1234, AirlineTest, test, test.

Airport	createAirlineManager	setupScenario3	airline: "Copa" id: 12345678 password: "1234" username: "AirlineTest" name: "test" lastname: "testL"	User created successfully. Returns Copa, 12345678, 1234, AirlineTest, test, test.
---------	----------------------	----------------	--	--

Crear nuevo usuario controlador torre de control

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en 2 casos, añadir correctamente el objeto al arreglo está vacío y cuando hay otros objetos en el.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario4	Airport	
setupScenario5	Airport	

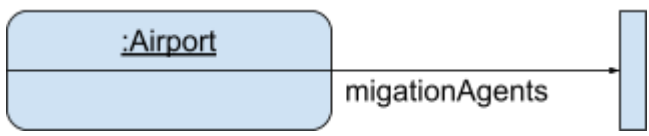
Diseño de Casos de Prueba

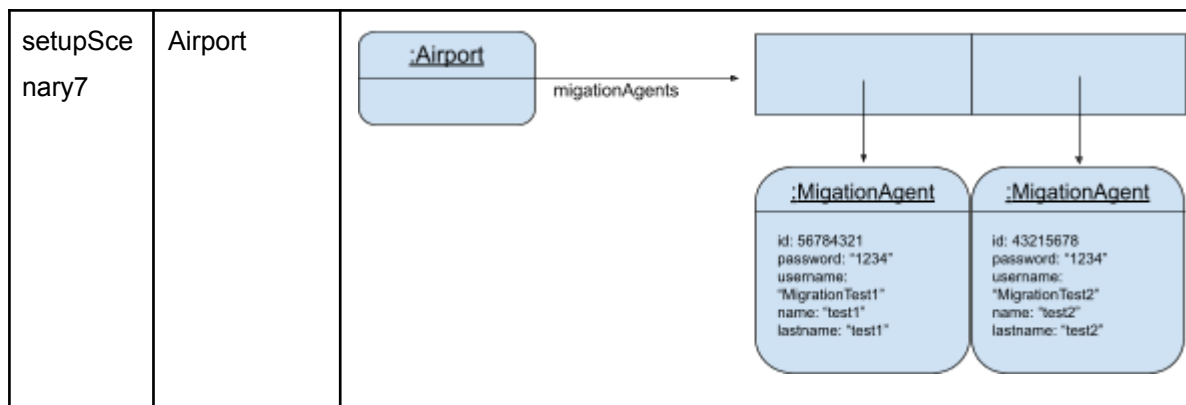
Objetivo de la Prueba: Registrar un nuevo controlador torre de control				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	createTowerCtrl	setupScenario4	id: 12345 password: "1234" username: "TowerTest" name: "test" lastname: "testL"	User created successfully. Returns 12345, 1234, TowerTest, test, testL.
Airport	createTowerCtrl	setupScenario5	id: 12345 password: "1234" username: "TowerTest" name: "test" lastname: "testL"	User created successfully. Returns 12345, 1234, TowerTest, test, testL.

Crear nuevo usuario agente de migración

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en 2 casos, añadir correctamente el objeto al arreglo está vacío y cuando hay otros objetos en el.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario6	Airport	 <pre> sequenceDiagram participant Airport as :Airport participant Agents as migrationAgents Airport->>Agents </pre>



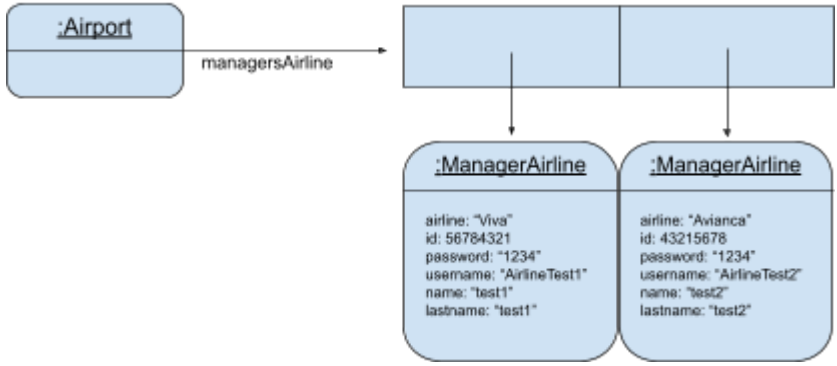
Diseño de Casos de Prueba

Objetivo de la Prueba: Registrar un nuevo agente de migración				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	createM igration Agent	setupScen ary6	id: 12345 password: "1234" username: "MigrationTest1" name: "test" lastname: "testL"	User created successfully. Returns 12345, 1234, MigrationTest1, test, testL.
Airport	createM igration Agent	setupScen ary7	id: 12345 password: "1234" username: "MigrationTest1" name: "test" lastname: "testL"	User created successfully. Returns 12345, 1234, MigrationTest1, test, testL.

Eliminar usuario administrador de aerolínea

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenary3	Airport	

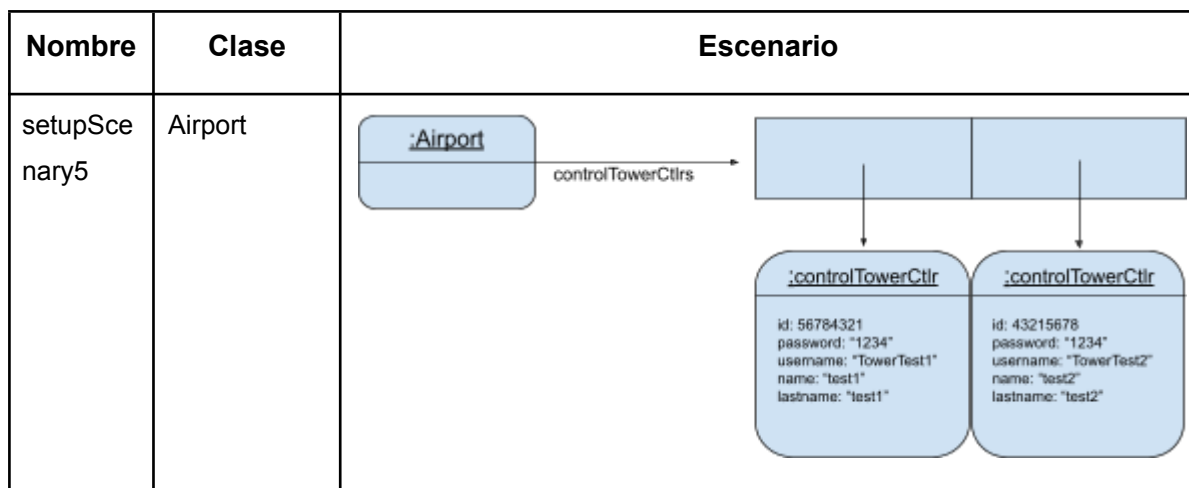
Diseño de Casos de Prueba

Objetivo de la Prueba: Eliminar administrador de aerolínea por Id.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	deletAirlineManager	setupScenary3	id: 56784321	User eliminated successfully. Returns array size = 1.

Eliminar usuario controlador torre de control

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios



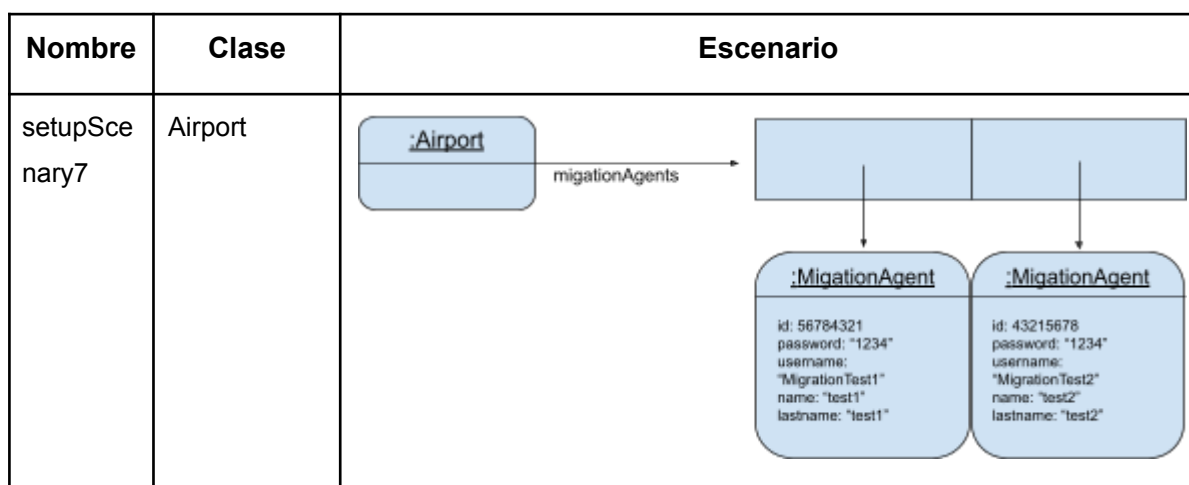
Diseño de Casos de Prueba

Objetivo de la Prueba: Eliminar controlador de torre por id				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	deleteTowerCtrl	setupScenario5	id: 56784321	User eliminated successfully. Returns array size = 1.

Eliminar usuario agente de migración

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios



Diseño de Casos de Prueba

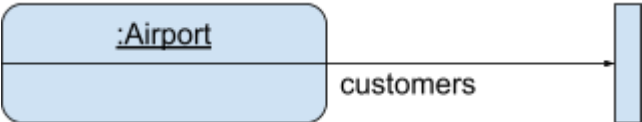
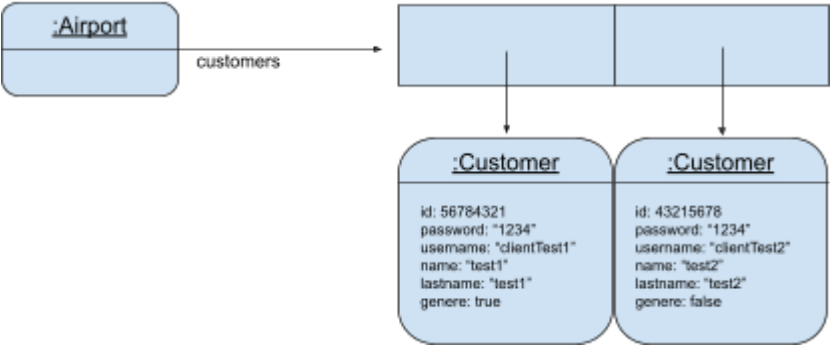
Objetivo de la Prueba: Eliminar agente de migración por id				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	deleteMigrationAgent	setupScenario7	id: 56784321	User eliminated successfully. Returns array size = 1.

Cliente Final - Casos

Crear nuevo usuario cliente

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en 2 casos, añadir correctamente el objeto al arreglo está vacío y cuando hay otros objetos en el.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario8	Airport	 <pre> sequenceDiagram participant Airport as :Airport Airport-->> customers: customers </pre>
setupScenario9	Airport	 <pre> sequenceDiagram participant Airport as :Airport Airport-->> customers: customers customers-->> Customer1: customers-->> Customer2: Customer1-->> Airport: id: 56784321, password: "1234", username: "clientTest1", name: "test1", lastname: "test1", genere: true Customer2-->> Airport: id: 43215678, password: "1234", username: "clientTest2", name: "test2", lastname: "test2", genere: false </pre>

Diseño de Casos de Prueba

Objetivo de la Prueba: Registrar un nuevo cliente.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	create Client	setupScen ary8	id: 12345 password: "1234" username: "clientTest" name: "test" lastname: "testL" genere: true	User created successfully. Returns 12345, 1234, clientTest, test, testL.
Airport	create Client	setupScen ary9	id: 12345 password: "1234" username: "clientTest" name: "test" lastname: "testL" genere: true	User created successfully. Returns 12345, 1234, clientTest, test, testL.

Eliminar usuario usuario cliente

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScen ary9	Airport	<pre> classDiagram class Airport { +customers } class Customer { +id +password +username +name +lastname +genere } Airport --> "2" Customer : customers </pre>

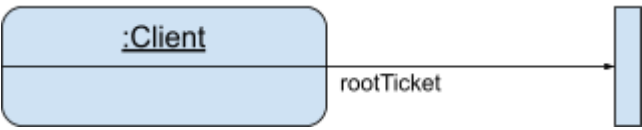
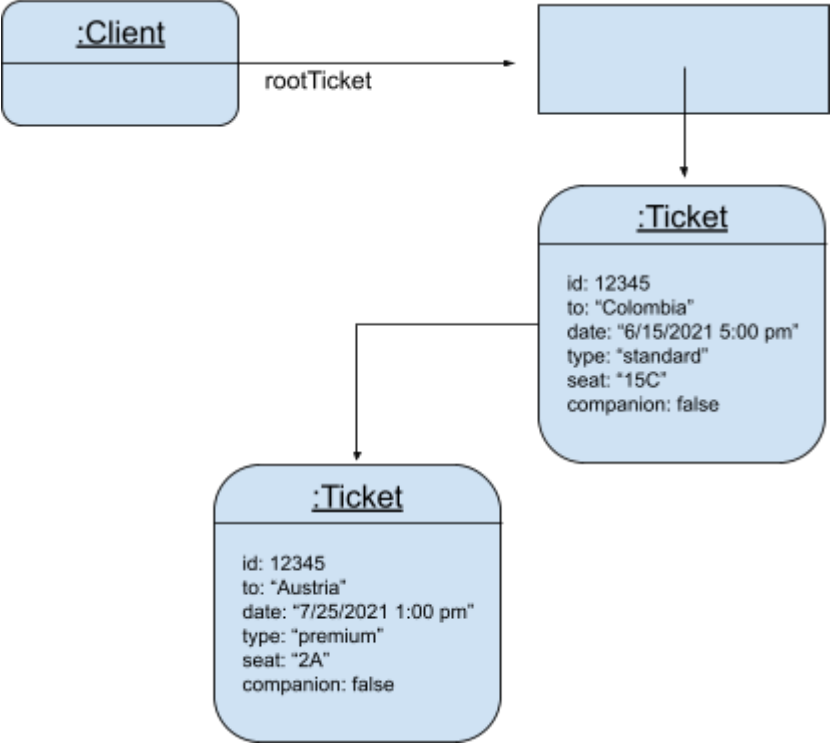
Diseño de Casos de Prueba

Objetivo de la Prueba: Eliminar cliente por Id.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	delete Client	setupScenario9	id: 56784321	User eliminated successfully. Returns array size = 1.

Crear nuevo ticket

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en 2 casos, añadir correctamente el objeto al arreglo está vacío y cuando hay otros objetos en el.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario1	Client	 <pre> sequenceDiagram participant Client as :Client Client->>[] : rootTicket </pre>
setupScenario2	Client	 <pre> sequenceDiagram participant Client as :Client Client->>[] : rootTicket activate [] []->>Ticket1 as :Ticket activate Ticket1 Ticket1->>Ticket2 as :Ticket activate Ticket2 Ticket2->>[] : deactivate Ticket2 deactivate Ticket1 deactivate [] </pre>

Diseño de Casos de Prueba

Objetivo de la Prueba: Crea un nuevo tiquete.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Client	createTicket	setupScenario1	id: 12345 to: "China" date: "8/5/2021 10:00 pm" type: "premium" seat: "1A" companion: false	Ticket created successfully. Returns 12345, China, 8/5/2021 10:00 pm, premium, 1A, false.
Client	createTicket	setupScenario2	id: 12345 to: "China" date: "8/5/2021 10:00 pm" type: "premium" seat: "1A" companion: false	Ticket created successfully. Returns 12345, China, 8/5/2021 10:00 pm, premium, 1A, false.

Eliminar ticket

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios

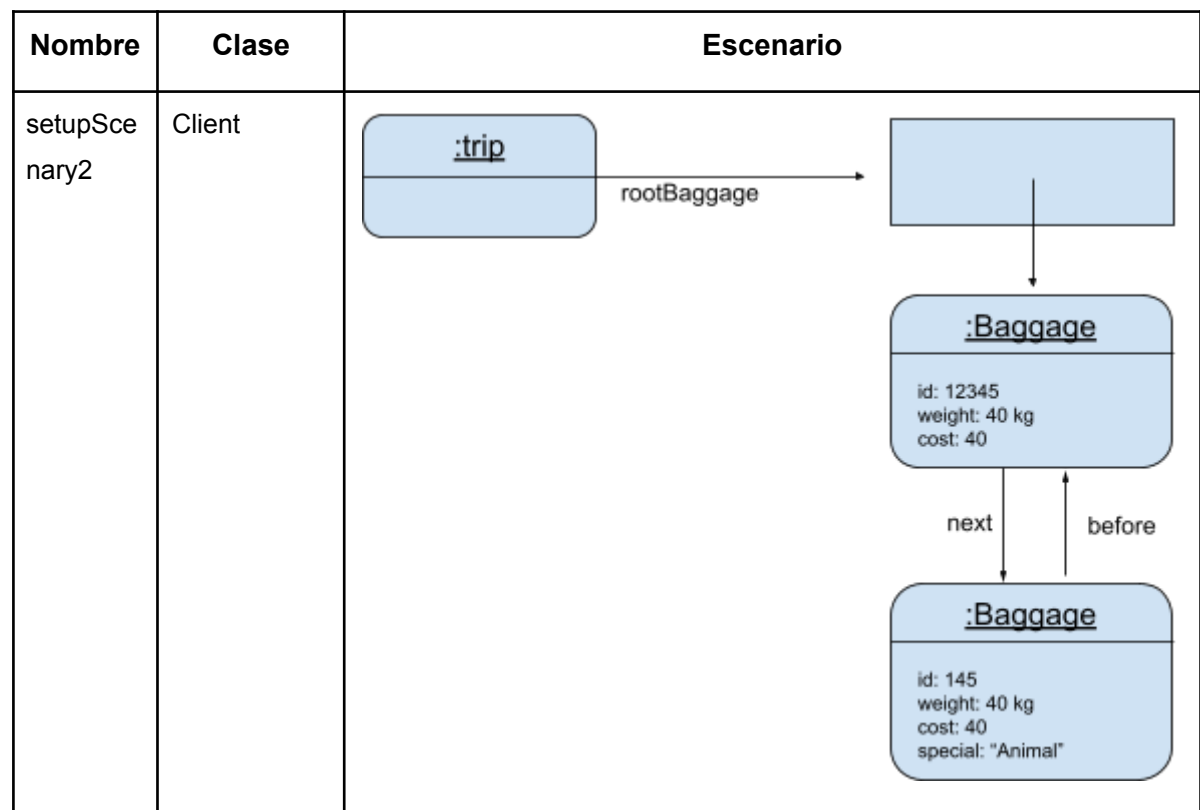
Nombre	Clase	Escenario
setupScenario2	Client	<pre> graph LR Client1[":Client"] -- tickets --> Container subgraph Container Ticket[":Ticket id: 12345 to: 'Colombia' date: '6/15/2021 5:00 pm' type: 'standard' seat: '15C' companion: false"] Client2[":Client id: 54321 to: 'Austria' date: '7/25/2021 1:00 pm' type: 'premium' seat: '2A' companion: false"] end </pre>

Diseño de Casos de Prueba

Objetivo de la Prueba: Eliminar tickete por Id.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Client	deletetic ket	setupScen ary11	id: 54321	Ticket eliminated successfully. Returns array size = 1.

Añadir equipaje

Configuración de los Escenarios



Diseño de Casos de Prueba

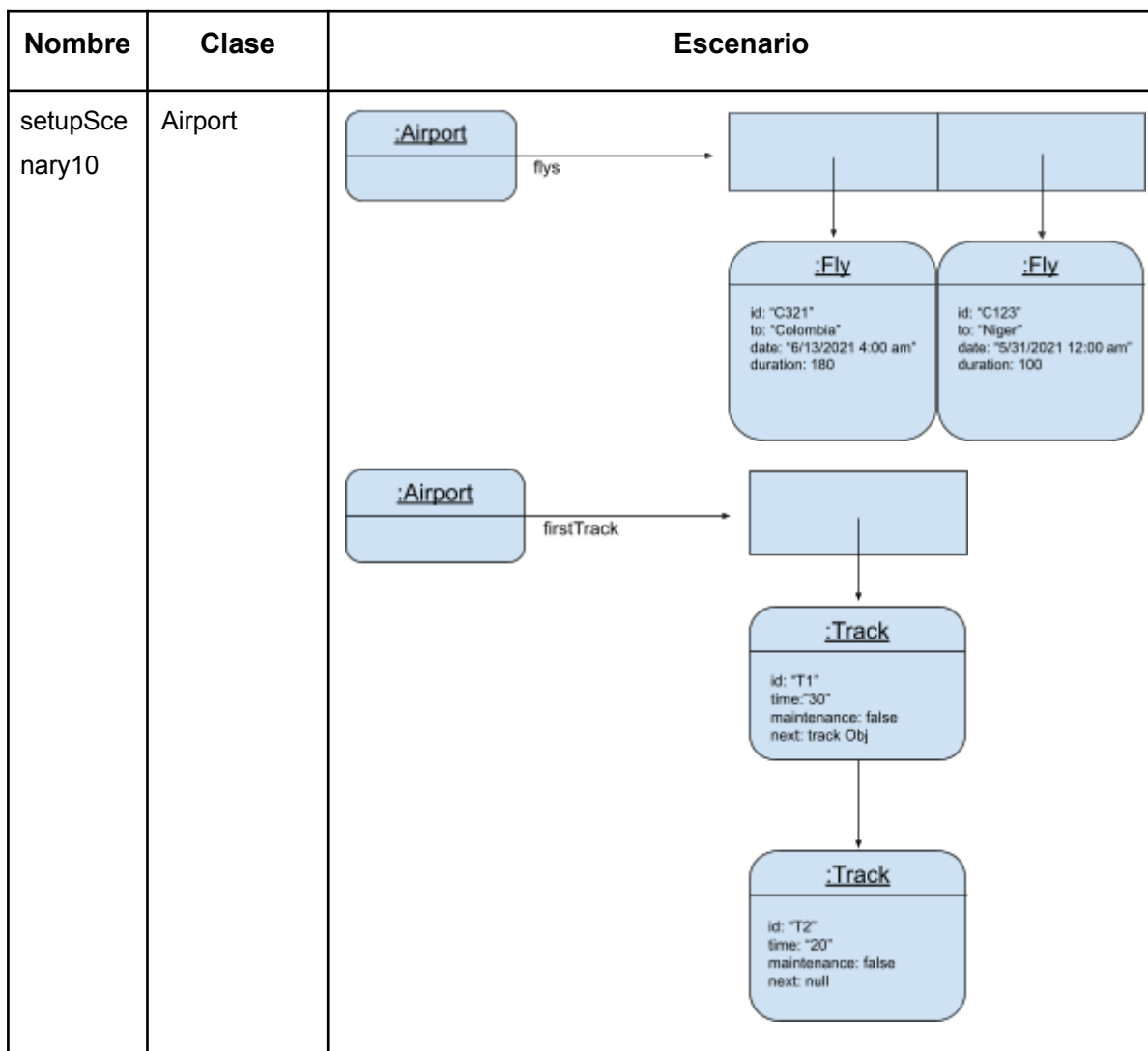
Objetivo de la Prueba: Añadir equipaje al vuelo				
Clase	Método	Escenario	Valores de Entrada	Resultado
Client	deletetic ket	setupScen ary11	id: 100 weight: 1	Baggage added successfully. Returns 100, 1, 100, gun.

			cost: 100 special: "gun"	
--	--	--	-----------------------------	--

Controlador torre de control - Casos

Asignar vuelo a pista

Configuración de los Escenarios

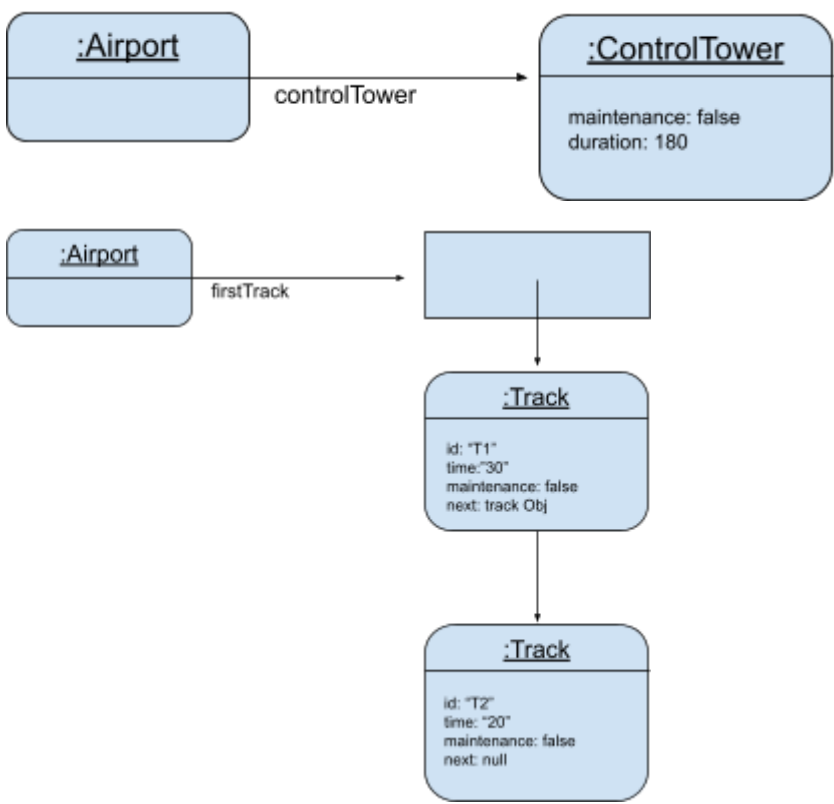


Diseño de Casos de Prueba

Objetivo de la Prueba: Asignar vuelo y una pista por un tiempo				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	assingFlyTrack	setupScenario10	id: C321 id: T1	true

Activar mantenimiento

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario11	Airport	 <pre>graph TD A["<u>:Airport</u>"] -- controlTower --> CT["<u>:ControlTower</u>
maintenance: false
duration: 180"] A -- firstTrack --> B[" "] B --> T1["<u>:Track</u>
id: 'T1'
time: '30'
maintenance: false
next: track Obj"] T1 --> T2["<u>:Track</u>
id: 'T2'
time: '20'
maintenance: false
next: null"]</pre>

Diseño de Casos de Prueba

Objetivo de la Prueba: Activar el período de mantenimiento				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	mainten ance	setupScen ary11		mantinance: true

Llamar lista de vuelos actuales

Configuración de los Escenarios

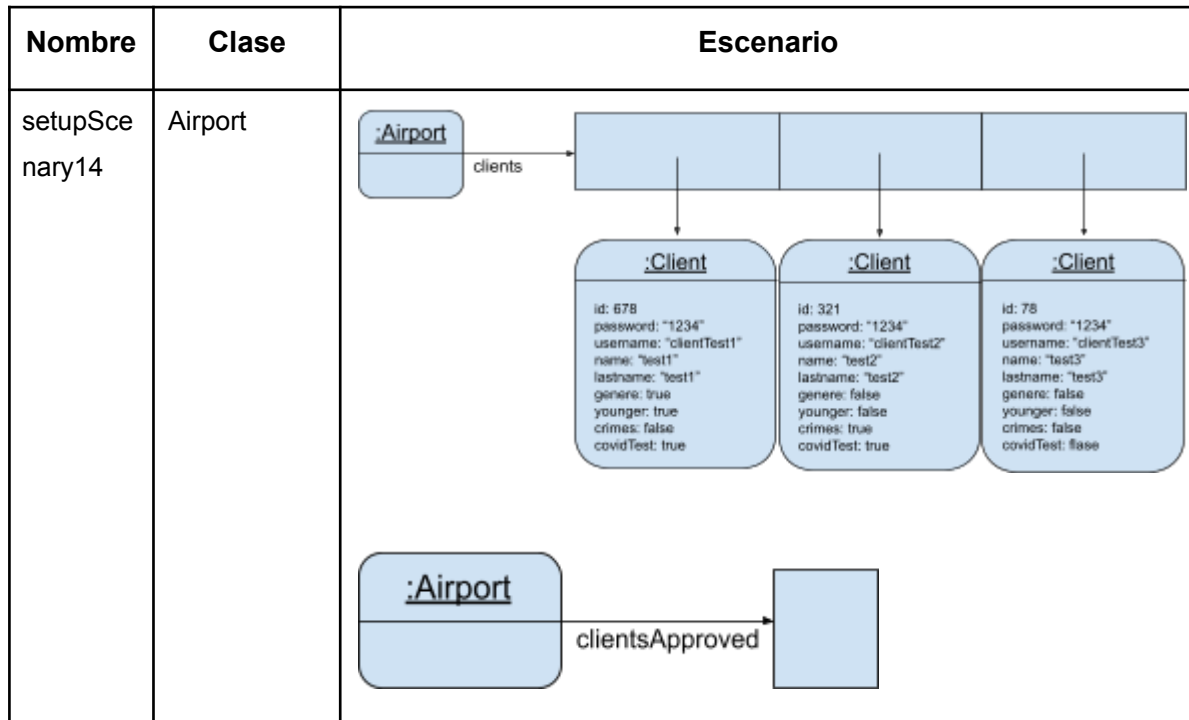
Nombre	Clase	Escenario
setupScenariy13	Airport	<pre> sequenceDiagram actor Airport as :Airport participant Boundary participant Fly1 as :Fly participant Fly2 as :Fly Airport->>Boundary: flys activate Boundary Boundary->>Fly1 activate Fly1 Fly1->>Fly2 deactivate Fly1 activate Fly2 Fly2->>Boundary deactivate Fly2 deactivate Boundary </pre>

Diseño de Casos de Prueba

Objetivo de la Prueba: Obtener lista de vuelos				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	getActualFlights	setupScenario13		id: C321 id: C123

Agente de migración - Casos

Aprobación de migración para pasar Configuración de los Escenarios



Diseño de Casos de Prueba

Objetivo de la Prueba: Validar la salida de un cliente				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airport	validateClientPass	setupScenario14	id: 678	fail clientsApproved size = 0
Airport	validateClientPass	setupScenario14	id: 321	fail clientsApproved size = 0


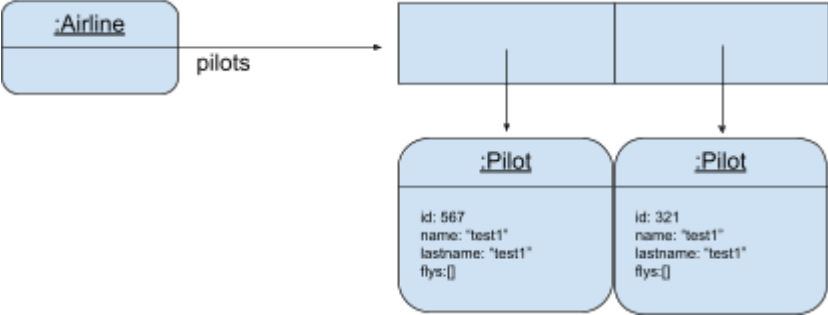
Airport	validat eClient Pass	setupScen ary14	id: 78	fail clientsApproved size = 0
---------	----------------------------	--------------------	--------	----------------------------------

Administrador de aerolínea - Casos

Crear nuevo piloto

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en 2 casos, añadir correctamente el objeto al arreglo está vacío y cuando hay otros objetos en el.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenary1	Airline	 <pre> classDiagram class Airline { <<name>> } class EmptyBox Airline --> EmptyBox : pilots </pre>
setupScenary2	Airline	 <pre> classDiagram class Airline { <<name>> } class Container { <<name>> } class Pilot { id name lastname flys } Airline --> Container : pilots Container --> Pilot1 Container --> Pilot2 class Pilot1 { id: 567 name: "test1" lastname: "test1" flys: [] } class Pilot2 { id: 321 name: "test1" lastname: "test1" flys: [] } </pre>

Diseño de Casos de Prueba

Objetivo de la Prueba: Crear un nuevo piloto de la aerolínea				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airline	create Pilot	setupScenary1	id: 123 name: "test" lastname: "testL" flys:[]	Created successfully. Returns 123, test, testL, [].
Airline	create Pilot	setupScenary2	id: 123 name: "test" lastname: "testL" flys:[]	Created successfully. Returns 123, test, testL, [].

Crear nuevo asistente

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en 2 casos, añadir correctamente el objeto al arreglo está vacío y cuando hay otros objetos en el.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenary3	Airline	<pre> classDiagram class Airline { +assistants } Airline --> [] : assistants </pre>
setupScenary4	Airline	<pre> classDiagram class Airline { +assistants } class Assistant { +id +name +lastname +flys } Airline --> Assistant : assistants Assistant --> Assistant : Assistant --> Assistant : </pre>

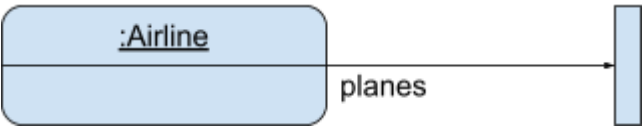
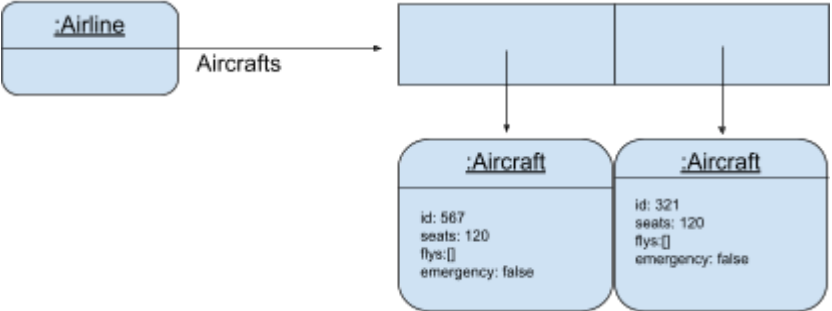
Diseño de Casos de Prueba

Objetivo de la Prueba: Crear un nuevo asistente de vuelo				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airline	createAssistant	setupScenario3	id: 123 name: "test" lastname: "testL" flys:[]	Created successfully. Returns 123, test, testL, [].
Airline	createAssistant	setupScenario4	id: 123 name: "test" lastname: "testL" flys:[]	Created successfully. Returns 123, test, testL, [].

Crear nuevo avión

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en 2 casos, añadir correctamente el objeto al arreglo está vacío y cuando hay otros objetos en el.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario5	Airline	 <pre> classDiagram class Airline { +planes } Airline --> [] : planes </pre>
setupScenario6	Airline	 <pre> classDiagram class Airline { +Aircrafts } class Aircraft { +id +seats +flys +emergency } Airline --> Aircraft : Aircrafts Aircraft --> Aircraft : </pre>

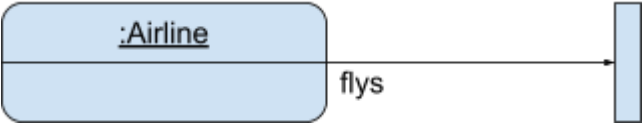
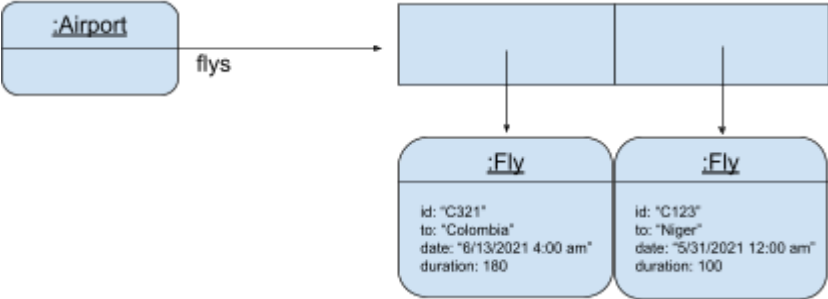
Diseño de Casos de Prueba

Objetivo de la Prueba: crear un nuevo avión				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airline	createPlane	setupScenario5	id: 123 seats: 150 flights: [] emergency: false	Created successfully. Returns 123, 150, [], false.
Airline	createPlane	setupScenario6	id: 123 seats: 150 flights: [] emergency: false	Created successfully. Returns 123, 150, [], false.

Crear nuevo vuelo

Se presenta a continuación el método encargado de **añadir una nueva instancia**, en 2 casos, añadir correctamente el objeto al arreglo está vacío y cuando hay otros objetos en el.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario7	Airline	
setupScenario8	Airline	

Diseño de Casos de Prueba

Objetivo de la Prueba: Crear un nuevo registro de vuelo en la aerolínea				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airline	createFly	setupScenario7	id: "C452" to: "Argentina" date: "8/2/2021 9:00 pm" duration: 280	Created successfully. Returns C452, Argentina, 8/2/2021 9:00 pm, 280.
Airline	createFly	setupScenario8	id: "C452" to: "Argentina" date: "8/2/2021 9:00 pm" duration: 280	Created successfully. Returns C452, Argentina, 8/2/2021 9:00 pm, 280.

Eliminar piloto

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario2	Airline	<pre> classDiagram class Airport { +pilots } class Pilot { +id +name +lastname +flys } Airport --> Pilot : pilots </pre>

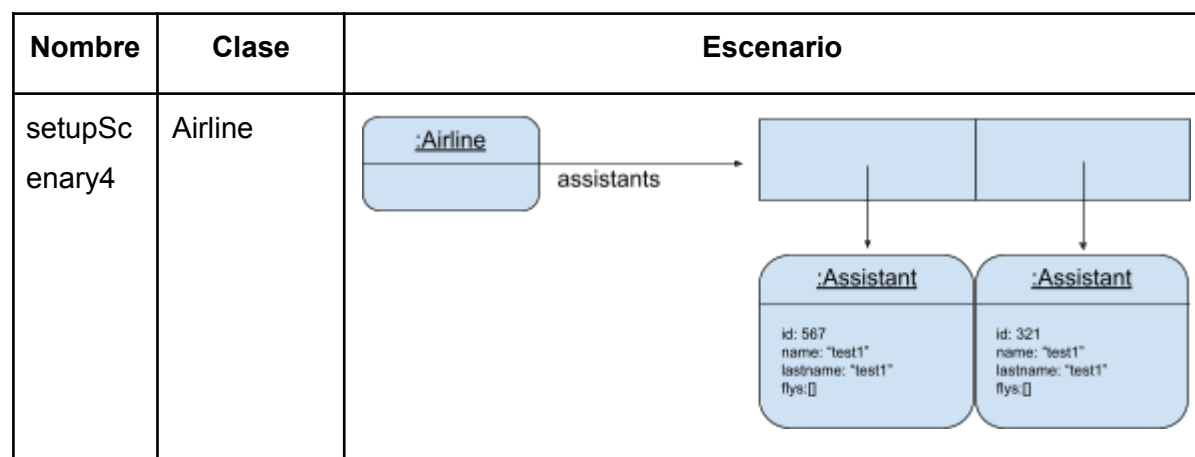
Diseño de Casos de Prueba

Objetivo de la Prueba: Eliminar un piloto por id				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airline	deletPlane	setupScenary2	id: 567	User eliminated successfully. Returns array size = 1.

Eliminar asistente

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios



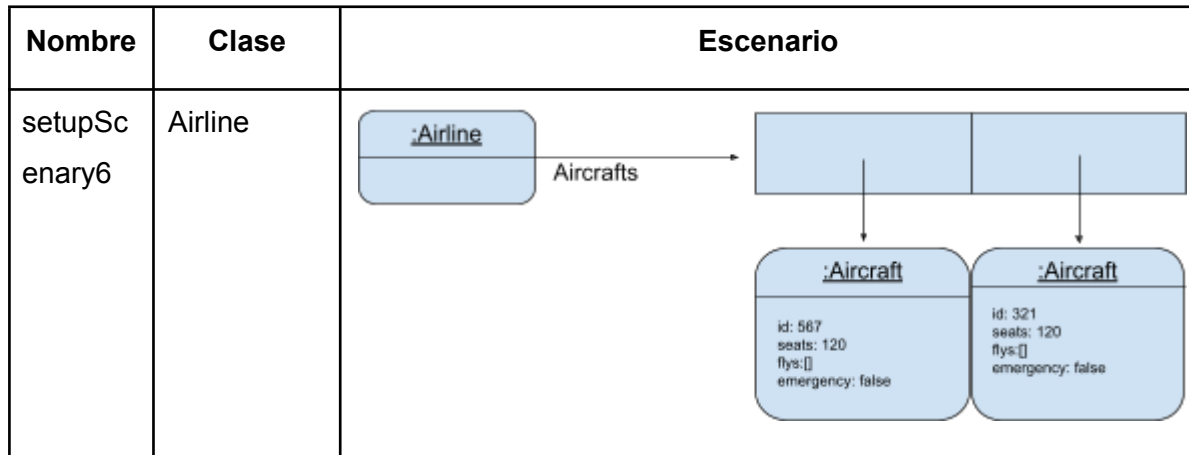
Diseño de Casos de Prueba

Objetivo de la Prueba: Eliminar asistente de vuelo por id				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airline	deletAsistant	setupScenary4	id: 321	User eliminated successfully. Returns array size = 1.

Eliminar avión

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios



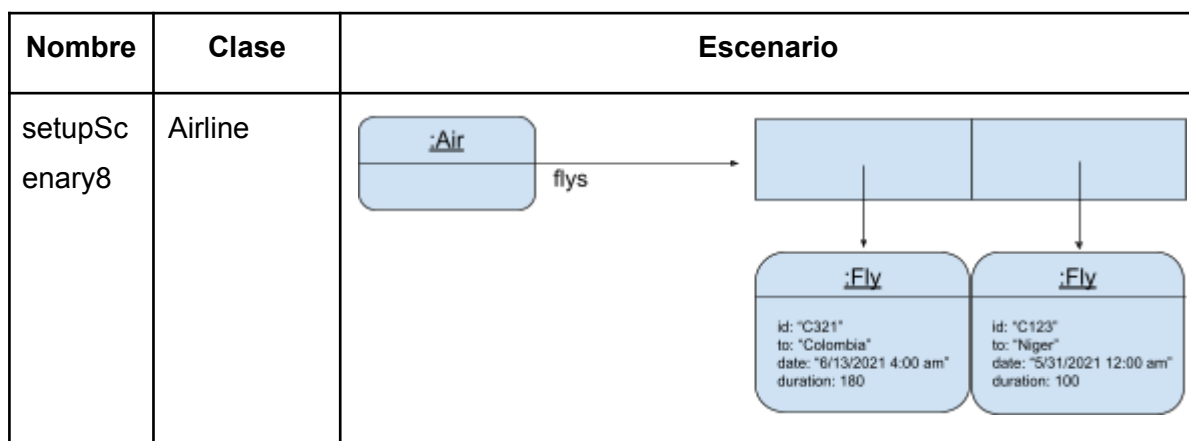
Diseño de Casos de Prueba

Objetivo de la Prueba: Eliminar un avión por id				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airline	deletePlane	setupScenario6	id: 567	User eliminated successfully. Returns array size = 1.

Eliminar vuelo

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios



Diseño de Casos de Prueba

Objetivo de la Prueba: Eliminar un vuelo por id				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airline	deleteFlight	setupScenario8	id: "C123"	User eliminated successfully. Returns array size = 1.

Asignar Asistentes a vuelo

Se presenta a continuación el método encargado de **eliminar una instancia**, en un caso, eliminar cuando hay otros objetos en el arreglo de instancias.

Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenario9	Airline	<pre>graph TD A["<u>:Aircraft</u>"] -- rootAssitent --> B[" "] B --> C["<u>:Aircraft</u>
id: 567
name: 'test1'
lastname: 'test1'
flys: []"] C --> D["<u>:Aircraft</u>
id: 321
name: 'test2'
lastname: 'test2'
flys: []"]</pre>

Diseño de Casos de Prueba

Objetivo de la Prueba: Añadir un asistente al vuelo				
Clase	Método	Escenario	Valores de Entrada	Resultado
Airline	deletPlane	setupScenary9	id: 678 name: "test" lastname: "testL" flys:[]	Assistant added successfully. Returns 678, test, testL, [].