**Escenarios**

|  |  |  |
| --- | --- | --- |
| Nombre | Clase | Escenario |
| setupScenary1 | FlightTest | airline = “Avianca”  time = "AM 10:30"  destination = "Colombia"  gate = 3  id = "ABC123"  date = "2010/10/10" |
| setupScenary2 | FlightTest | A las dos relaciones de Flight, se les asigna los siguiente:  f = new Flight("Avianca", "AM 10:30", "Colombia", 3, "ABC123", "2010/10/10")  f2 = new Flight("Avianca", "AM 10:30", "Colombia", 3, "ABC124", "2010/10/10") |
| setupScenary1 | AirportTest |  |
| setupScenary2 | AirportTest | A la relación de airport (ap) le asigno un objeto de Airport, pasando 20 por parámetro, para completar el constructor.  (ap = new Airport(20)) |
| setupScenary3 | AirportTest | A la lista de Flights que tiene Airport (ap), le asigno una lista aleatoria usando el método randomFlightList(10). Pasandole 10 como parámetro para crear 10 objetos de tipo Flight y meterlos a esa lista. Los valores de los atributos de cada Flight son aleatorios. |

**Pruebas Unitarias**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el constructor y los getters de la clase Flight funcionen correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Flight | Flight | setupScenary1 |  | Se crea un objeto de Flight con las entradas correspondientes correctamente. Al usar el get de cada atributo, se devuelve el valor correcto. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el método compareTo funciona correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Flight | compareTo | setupScenary2 |  | Al comparar ambos vuelos, ya que el que llama al método es menor, devuelve un número menor a 0. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el método time funciona correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Airpot | time | setupScenary2 | Double num | Se espera que retorne un String que contenga “AM” si num es mayor o igual a 100.0 y se espera que retorne un String que contenga “PM” si num es menor a 100.0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el método sortByGate funciona correctamente. | | | | | | | | | | | | | | |
| **Clase** | **Método** | | | | | **Escenario** | | | | **Valores de Entrada** | | | | **Resultado** |
| Airport | sortByGate | | | | | setupScenary2  setupScenary3 | | | |  | | | | Se espera que ahora la lista esté organizada de menor a mayor por sus respectivas puertas. |
| **Objetivo de la Prueba:** Comprobar que el método sortByDate funciona correctamente. | | | | | | | | | | | | | | |
| **Clase** | **Método** | | | | | **Escenario** | | | | **Valores de Entrada** | | | | **Resultado** |
| Airport | sortByDate | | | | | setupScenary2  setupScenary3 | | | |  | | | | Se espera que ahora la lista esté organizada de menor a mayor por sus respectivas fechas. |
| **Objetivo de la Prueba:** Comprobar que el método sortById funciona correctamente. | | | | | | | | | | | | | | |
| **Clase** | **Método** | | | | **Escenario** | | | | **Valores de Entrada** | | | | **Resultado** | |
| Airport | sortById | | | | setupScenary2  setupScenary3 | | | |  | | | | Se espera que ahora la lista esté organizada de menor a mayor por sus respectivos identificadores de vuelo. | |
| **Objetivo de la Prueba:** Comprobar que el método sortByTime funciona correctamente. | | | | | | | | | | | | | | |
| **Clase** | **Método** | | | | **Escenario** | | | | **Valores de Entrada** | | | | **Resultado** | |
| Airport | sortByTime | | | | setupScenary2  setupScenary3 | | | |  | | | | Se espera que ahora la lista esté organizada de menor a mayor por sus respectivos horarios de salida. | |
| **Objetivo de la Prueba:** Comprobar que el método sortByAirline funciona correctamente. | | | | | | | | | | | | | | |
| **Clase** | **Método** | | | **Escenario** | | | | **Valores de Entrada** | | | | **Resultado** | | |
| Airport | sortByAirline | | | setupScenary2  setupScenary3 | | | |  | | | | Se espera que ahora la lista esté organizada de menor a mayor por sus respectivas aerolíneas. | | |
| **Objetivo de la Prueba:** Comprobar que el método sortByDestination funciona correctamente. | | | | | | | | | | | | | | |
| **Clase** | | **Método** | **Escenario** | | | | **Valores de Entrada** | | | | **Resultado** | | | |
| Airport | | sortByDestination | setupScenary2  setupScenary3 | | | |  | | | | Se espera que ahora la lista esté organizada de menor a mayor por sus respectivos destinos. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el método searchFlightAirline funciona correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Airport | searchFlightAirline | setupScenary2  setupScenary3 | String airline | Se espera que, de no devolver nulo (que no lo encontró en la lista), el método devuelva el primer Flight que corresponda con el criterio de búsqueda. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el método searchFlightTime funciona correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Airport | searchFlightTime | setupScenary2  setupScenary3 | String time | Se espera que, de no devolver nulo (que no lo encontró en la lista), el método devuelva el primer Flight que corresponda con el criterio de búsqueda. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el método searchFlightDate funciona correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Airport | searchFlightDate | setupScenary2  setupScenary3 | String date | Se espera que, de no devolver nulo (que no lo encontró en la lista), el método devuelva el primer Flight que corresponda con el criterio de búsqueda. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el método searchFlightId funciona correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Airport | searchFlightId | setupScenary2  setupScenary3 | String id | Se espera que, de no devolver nulo (que no lo encontró en la lista), el método devuelva el primer Flight que corresponda con el criterio de búsqueda. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el método searchFlightDestination funciona correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Airport | searchFlightDestination | setupScenary2  setupScenary3 | String destination | Se espera que, de no devolver nulo (que no lo encontró en la lista), el método devuelva el primer Flight que corresponda con el criterio de búsqueda. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Comprobar que el método searchFlightGate funciona correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Airport | searchFlightGate | setupScenary2  setupScenary3 | int gate | Se espera que, de no devolver nulo (que no lo encontró en la lista), el método devuelva el primer Flight que corresponda con el criterio de búsqueda. |