1. Requerimientos Funcionales

|  |  |
| --- | --- |
| **Nombre:** | R. #1. Permitir mostrar vuelos |
| **Resumen:** | Permite mostrar la información de los vuelos |
| Entradas: |  |
|  |  |
| Resultados: | Lista de vuelos |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #2. Mostrar hora en formato no militar |
| **Resumen:** | Permite ver la hora de los vuelos en formato no militar |
| Entradas: | Ninguno |
|  |  |
| Resultados: | Hora no militar |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #3. Generar listado de vuelos |
| **Resumen:** | Permite generar un listado de vuelos con fechas, horarios, aerolíneas, números de vuelos totalmente aleatorios (único), ciudades de destino y puertas de embarque. |
| Entradas: | Cantidad de vuelos |
|  |  |
| Resultados: | Lista de vuelos |
|  |  |

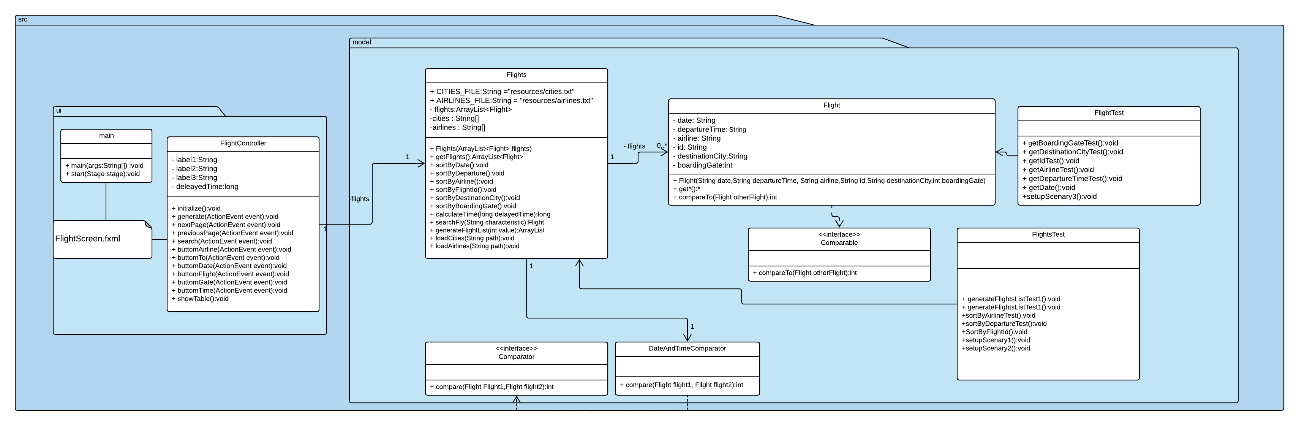
|  |  |
| --- | --- |
| **Nombre:** | R. #4. Permitir paginar los vuelos |
| **Resumen:** | Permite si la cantidad de vuelos es mayor al espacio en pagina poder navegar entre paginas con un next y previous |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Navegar entre paginas |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #5. Permitir ordenar los vuelos |
| **Resumen:** | Permitir ordenar los vuelos según cualquier criterio: Hora, fecha, vuelo, lugar, etc |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Lista de vuelos ordenados |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | R. #6. Permitir buscar un vuelo |
| **Resumen:** | Permitir buscar vuelos por cualquier criterio y si hay más de uno mostrar el primer criterio encontrado |
| Entradas: | Fecha, hora, vuelo, aerolinia, etc. |
|  |  |
| Resultados: | Si: Dar primer vuelo encontrado |
|  | No: Dar vuelo encontrado |

|  |  |
| --- | --- |
| **Nombre:** | R. #. Permitir dar tiempo |
| **Resumen:** | Permite dar tiempo en que se tardo en hacer una búsqueda |
| Entradas: | Ninguna |
|  |  |
| Resultados: | Ninguna |
|  |  |

1. Diagrama de Clases



1. Casos de prueba

Diseño de escenarios

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setupScenary1 | FlightsTest | Se crea el escenario con un nueva lista de tipo Flights para pruebas correctas de la clase Flights |
| setupScenary2 | FlightsTest | Se crea el escenario vacío para que falle con excepción de NullPointerException |
| setupScenary3 | FlightTest | Se crea el escenario con un vuelo creado con atributos por defecto |

Diseño de las pruebas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Flights | generateFlightList | setupScenary1 | value = 2 | True  Permite crear una lista de 2 vuelos con sus especificaciones aleatorias |
| Flights | generateFlightList | setUpScenary2 | Value = 2 | False No permite crear una lista de 2 vuelos porque tira una excepción NullPointerException |
| Flights | sortByDeparture | setupScenary1 | Value = 2 | True Permite ordenar correctamente los dos vuelos de la lista creada según la salida del vuelo |
| Flights | sortByAirline | setupScenary1 | Value = 2 | True  Permite ordenar correctamente los dos vuelos de la lista creada según la aerolínea |
| Flights | sortByFlightId | setupScenary1 | Value = 2 | True  Permite ordenar correctamente los dos vuelos de la lista creada según el ID del vuelo |
| Flight | getBoardingGate | setupScenary3 | boardingGate = 5 | True Permite obtener el dato correctamente |
| Flight | getDestinationCity | setupScenary3 | destinationCity = “Cali” | True Permite obtener el dato correctamente |
| Flight | getId | setupScenary3 | Id = “1642” | True  Permite obtener el dato correctamente |
| Flight | getAirline | setupScenary3 | Airline = “DELTA” | True Permite obtener el dato correctamente |
| Flight | getDepartureTime | setupScenary3 | departureTime = “01:00 PM” | True  Permite obtener el dato correctamente |
| Flight | getDate | setupScenary3 | Date = “2019-02-19” | True Permite obtener el dato correctamente |