**Requerimientos funcionales**

|  |  |
| --- | --- |
| Nombre: | R1. Mostar vuelos |
| Descripción: | Se debe mostrar en pantalla toda la información relacionada con un vuelo, y la hora se debe presentar en formato 12 horas AM/PM.  Se debe mostrar en pantalla un máximo de 10 vuelos, por lo que, si el usuario crea más, estas deben mostrar en varias páginas, de 10 elementos |
| Entradas: |  |
| Salidas: | Se muestra en la ventana del programa la información de todos los vuelos, 10 por página. |

|  |  |
| --- | --- |
| Nombre: | R2. Generar aleatoriamente un listado de vuelos |
| Descripción: | El programa genera un listado de vuelos, en diferentes fechas, horarios, aerolíneas, números de vuelo (Identificador único), ciudad de destino y puertas de embarque.  Se genera un nuevo listado cada vez que el usuario lo desee. |
| Entradas: | Entero con la cantidad de vuelos a generar |
| Salidas: |  |

|  |  |
| --- | --- |
| Nombre: | R3. Ordenar vuelos por fecha y hora de salida |
| Descripción: | El programa los vuelos que se tienen por fecha y hora de salida, de menor a mayor, justo después de haberlos generado. |
| Entradas: | Lista con los vuelos generados |
| Salidas: | Lista con los vuelos ordenados por fecha y hora de salida de menor a mayor |

|  |  |
| --- | --- |
| Nombre: | R4. Buscar vuelo |
| Descripción: | El programa debe permitir buscar un vuelo entre todos los disponibles, mostrando en pantalla únicamente el buscado. Esta búsqueda puede ser por cualquier criterio. |
| Entradas: | Número de vuelo o fecha, hora, ciudad, puerta de embarque. |
| Salidas: | Se muestra en pantalla únicamente el primer vuelo encontrado, con toda su información relacionada. |

|  |  |
| --- | --- |
| Nombre: | R5. Ordenar vuelos según criterio del usuario |
| Descripción: | El programa permite ordenar los vuelos por cualquier criterio elegido por el usuario |
| Entradas: | ActionEvent |
| Salidas: | Lista con los vuelos ordenados según criterio del usuario |

|  |  |
| --- | --- |
| Nombre: | R6. Navegar entre paginas |
| Descripción: | El programa permite navegar entre las distintas páginas del programa, hacia adelante o hacia atrás |
| Entradas: | ActionEvent |
| Salidas: | Se muestra en pantalla la página seleccionada por el usuario, con sus vuelos correspondientes |

|  |  |
| --- | --- |
| Nombre: | R7. Mostar tiempo que tardó el programa realizando una búsqueda |
| Descripción: | El programa muestra en pantalla, el tiempo que tardó en realizar una búsqueda u ordenamiento. |
| Entradas: |  |
| Salidas: | Long con el tiempo tomado |

**Trazabilidad**

|  |  |  |
| --- | --- | --- |
| Requerimiento | Método | Clase |
| R1 | showTable() | FlightsController |
| R2 | generateFlightsList(int value) | Flights |
| R3 | sortByDate() | Flghts |
| R4 | searchFly() | Flights |
| R5 | sortByDate()  sortByDeparture()  sortByAirline()  sortByDestinationCity()  sortByBoardingGate()  sortByFlightId() | Flights |
| R6 | nextPage()  previousPage() | FlightsController |
| R7 | calculateTime() | Flgihts |

**Casos de prueba**

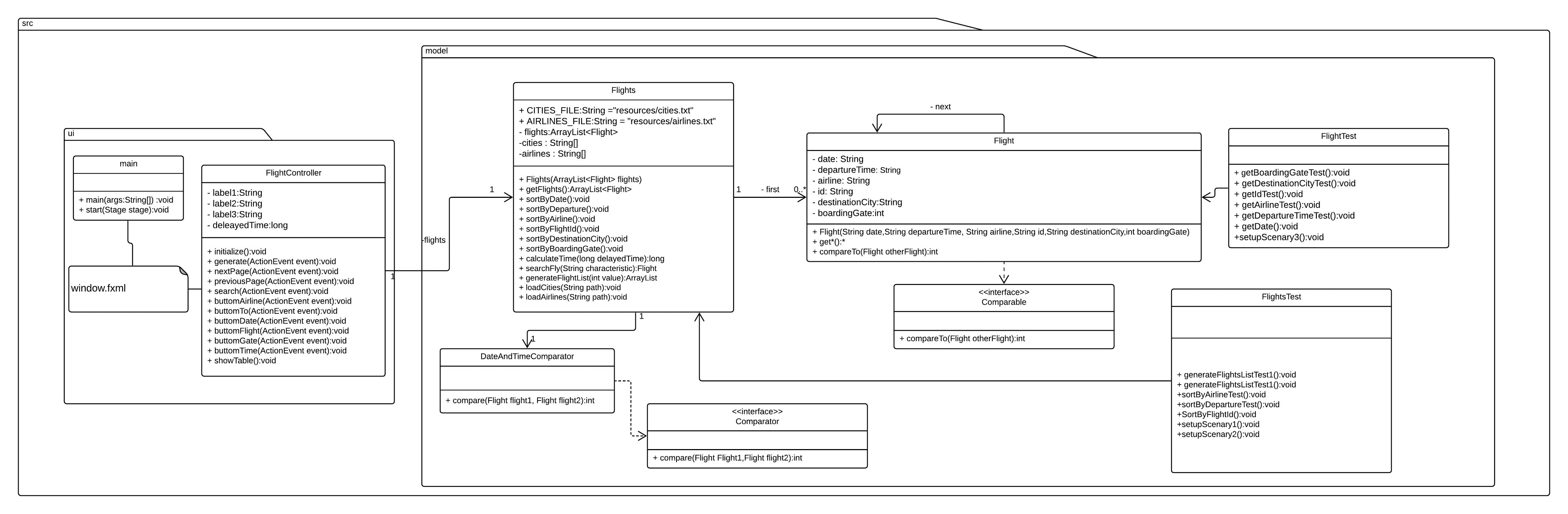
**Configuración de los Escenarios**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setupScenary1() | Flights | Relacion con la clase Flight de tipo ArrayList<Flight> |
| setupScenary2() | Flight | Objeto de tipo Flight, con los siguientes valores:   * Date: “2019-01-31” * departureTime=”12:00 AM” * airline=”Avianca” * id=”8742” * destinationCity=”Braga” * boardingGate=5 |

**Diseño de Casos de Prueba**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba: Verificar que se ha creado correctamente un vuelo** | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Flight | getBoardingGate() | setupScenary2() |  | True  Se tomo correctamente la puerta de entrada del vuelo |
| Flight | getDestinationCity() | setupScenary2() |  | True  Se tomo correctamente la puerta de entrada del vuelo |
| Flight | getId() | setupScenary2() |  | True  Se tomo correctamente la puerta de entrada del vuelo |
| Flight | getAirline() | setupScenary2() |  | True  Se tomo correctamente la puerta de entrada del vuelo |
| Flight | getDepartureTime() | setupScenary2() |  | True  Se tomo correctamente la puerta de entrada del vuelo |
| Flight | getDate() | setupScenary2() |  | True  Se tomo correctamente la puerta de entrada del vuelo |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

DIAGRAMA DE CLASES