LABORATORIO 4

1. Requerimientos funcionales

|  |
| --- |
| RF #1. Generar un listado de n vuelos. |
| Resumen: El programa genera aleatoriamente un listado de vuelos organizados por defecto con la hora de salida de menor a mayor, mostrando diferentes fechas, horarios, diferentes aerolíneas, números de vuelo, ciudades de destino y puertas de embarque.   * El programa permite al usuario decidir cuántos vuelos se van a generar. Si el número de vuelos excede el límite por pantalla, se procede a mostrar los siguientes n vuelos en otra página. |
| Entradas: El número de vuelos a mostrar. |
| Salidas: El programa muestra el listado de vuelos junto a todas las características correspondientes a cada uno, de tamaño escogido por el usuario. |

|  |
| --- |
| RF #2. El usuario decide el ordenamiento de los vuelos. |
| Resumen: El programa permite al usuario la posibilidad de ordenar los vuelos por cualquiera de los otros criterios: fecha, número de vuelo, ciudad de destino, etc. |
| Entradas: No tiene entradas. |
| Salidas: El programa muestra la nueva lista generada por el usuario. |

|  |
| --- |
| RF #3. El usuario puede realizar una búsqueda de un vuelo. |
| Resumen: El programa permite al usuario la posibilidad de buscar algún vuelo por cualquiera de los otros criterios: fecha, número de vuelo, ciudad de destino, etc. Si hay más de un vuelo que concuerde con el criterio buscado, entonces se mostrará el primer vuelo encontrado. |
| Entradas: No tiene entradas. |
| Salidas: El programa muestra los vuelos encontrados de acuerdo al criterio seleccionado. |

|  |
| --- |
| RF #4. El programa muestra el tiempo de búsqueda u ordenamiento. |
| Resumen: El programa muestra en pantalla la cantidad de tiempo en que tarda en realizar una operación de búsqueda u ordenamiento cada vez que el usuario decide alguna de esas dos acciones. |
| Entradas: No tiene entradas. |
| Salidas: El programa muestra el tiempo que tardó en generar la acción. |

1. Trazabilidad.

|  |  |
| --- | --- |
| Requerimiento | Método que lo soluciona |
| Generar un listado de n vuelos. | Generate(ActionEvent event)  getRandomDate()  getRandomAirline()  getRandomDestiny()  randomChar()  getRandomCode()  showTable()  clearData()  showTable()  nextPageGo(ActionEvent event)  lastPageGo(ActionEvent event)  clearData() |
| El usuario decide el ordenamiento de los vuelos. | orderByAirline(ActionEvent event)  orderByDate(ActionEvent event)  orderByDestiny(ActionEvent event)  orderByFlight(ActionEvent event)  orderByGate(ActionEvent event)  sortByAirline()  sortByFlight()  sortByDestiny()  sortByGate()  insertionSort()  bubbleSort()  selectionSort() |
| El usuario puede realizar una búsqueda de un vuelo. | binarySort(String parameter, String value)  linealSort(String parameter, String value)  configureComboBox()  searchAction(ActionEvent event)  choiceSearch(ActionEvent event) |
| El programa muestra el tiempo de búsqueda u ordenamiento. | generate(ActionEvent event) |

1. Diagrama de clases

