

<b>Nombre</b>	<b>R.F# 1.</b>
<b>Resumen</b>	Se hace una implementación básica de ordenamiento
<b>Entradas</b>	
<b>Resultados</b>	
Vuelos en un orden.	

<b>Nombre</b>	<b>R.F# 2.</b>
<b>Resumen</b>	Implementar métodos de búsqueda secuencial
<b>Entradas</b>	
Nombre de la Aerolínea (String)	
<b>Resultados</b>	
Posición del objeto en un arreglo ordenado	

<b>Nombre</b>	<b>R.F# 3.</b>
<b>Resumen</b>	Búsqueda binaria
<b>Entradas</b>	
Numero de vuelo único (int)	
<b>Resultados</b>	
Posición del objeto en un arreglo ordenado	

<b>Nombre</b>	<b>R.F# 4.</b>
<b>Resumen</b>	Implementar interfaz comparable

<b>Entradas</b>
Flight
<b>Resultados</b>
-1, 0, 1

<b>Nombre</b>	<b>R.F# 5.</b>
<b>Resumen</b>	Implementar interfaz Comparator
<b>Entradas</b>	
Flight, Flight	
<b>Resultados</b>	
-1, 0, 1	

<b>Nombre</b>	<b>R.F# 6.</b>
<b>Resumen</b>	Generar de manera aleatoria la lista de los vuelos posibles
<b>Entradas</b>	
Numero de vuelos que se generan (int)	
<b>Resultados</b>	
Se genera una nueva lista de forma aleatoria	

## Trazabilidad del Análisis al Diseño

Requerimientos Funcionales	Método	Clase
1	Main()  sortByAirline, sortByDate, sortByDestiny, sortByDestiny, sortByFlightNumber ,sortByGate,sortByHour,generateFlightInfo	Main  ScreenController
3	Main()  findFlight,generateFlightInfo	Main  ScreenController
2	Main()  findFlight,generateFlightInfo	Main  ScreenController
4	Main()  sortByAirline	Main  ScreenController
5	Main()  sortByDestiny	Main  ScreenController
6	Main()  initialize()	Main  ScreenController

