

DOCUMENTACION

LABORATORIO 5

Requerimientos funcionales:

Nombre	R0: Generar la hora en formato no militar
Resumen	Para evitar las confusiones con la hora militar, el tiempo se despliega en la forma horas : minutos (am/pm)
Entrada	N/A
Salida	Hora en formato de 12 horas

Nombre	R1: Generar vuelos con valores aleatorios
Resumen	Cada vez que se oprime el botón de generar, se crean la cantidad indicada de vuelos con atributos aleatorios
Entrada	ActionEvent
Salida	ArrayList<Plane> con todos los vuelos generados

Nombre	R2: Desplegar los vuelos
Resumen	La información de los vuelos generados se despliegan en un módulo ListView<String>
Entrada	N/A
Salida	Panel con la información de todos los vuelos generados

Nombre	R3: Ordenar los vuelos según sus atributos
Resumen	Se ordenan los vuelos según sus atributos, horario, fecha, destino, puerta, id y aerolínea
Entrada	ActionEvent
Salida	Un ArrayList con los vuelos ordenados según el atributo elegido

Nombre	R4: Buscar un vuelo específico según cualquier atributo
Resumen	Se busca un vuelo específico según el atributo que elija el usuario
Entrada	ActionEvent, Atributo : String
Salida	Ventana emergente con la información del vuelo solicitado

Nombre	R5: Desplegar el tiempo demorado en ejecutar una búsqueda u ordenamiento
Resumen	Para cada operación de búsqueda y ordenamiento, se imprime el tiempo que tomo para que este se ejecutara
Entrada	N/A
Salida	Tiempo que se demora el método en milisegundos.

Nombre	R6: desplegar la fecha en formato AAAA-MM-DD
Resumen	Se despliega la fecha de los vuelos en el formato AAAA-MM-DD
Entrada	N/A
Salida	La fecha de los vuelos se muestra de la forma AAAA-MM-DD

Nombre	R6: Almacenar los vuelos en una lista enlazada
Resumen	En vez de utilizar una matriz o lista, se almacenan los vuelos en una lista enlazada sencilla
Entrada	N/A
Salida	Los vuelos se enlazan de manera sencilla, no doble

Diagrama de clases:

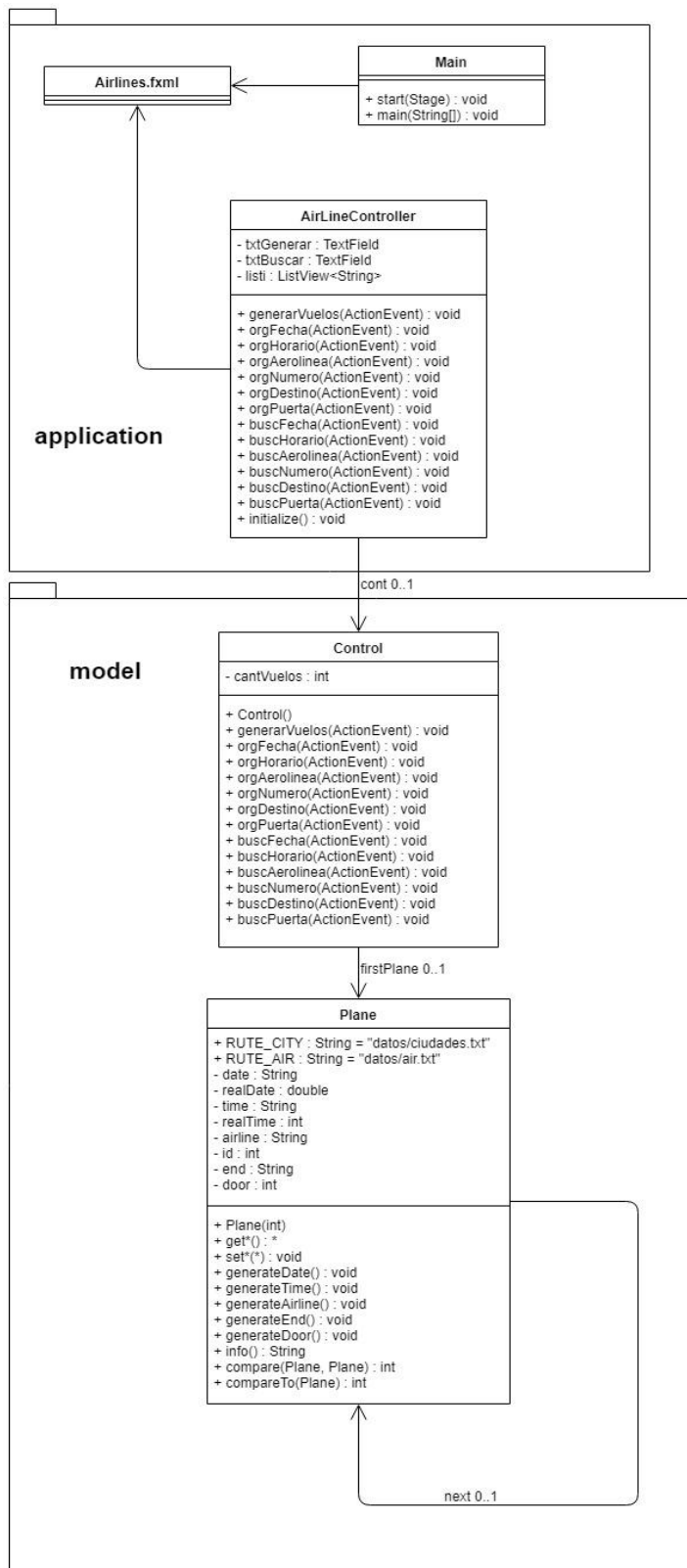


Tabla de trazabilidad del análisis al diseño:

Requerimiento	Clase	Metodo
Generar la hora en formato no militar	Plane	GenerateTime() : void
Generar vuelos con valores aleatorios	AirlineController	GenerarVuelos(ActionEvent) : void
	Control	GenerarVuelos(String) : void
	Plane	Generate*() : void
Desplegar los vuelos	AirlineController	GenerarVuelos(ActionEvent) : void
	Control	GenerarVuelos(String) : void
	Plane	Info() : String
Ordenar los vuelos según sus atributos	AirlineController	Org*((ActionEvent) : void
	Control	Org*(String) : void
	Plane	Get*() : *
Buscar un vuelo específico según cualquier atributo	AirlineController	Busc*((ActionEvent) : void
	Control	Busc*(String) : void
	Plane	Get*() : *
Desplegar el tiempo demorado en ejecutar una búsqueda u ordenamiento	AirlineController	Busc*((ActionEvent) : void Org*((ActionEvent) : void
desplegar la fecha en formato AAAA-MM-DD	Plane	GenerateDate() : void
Almacenar los vuelos en una lista enlazada	Plane	GetNext() : Plane setNext(Plane) : void

Diagrama de clases de pruebas unitarias:

