



Objetivos del programa

El proyecto de gestión aeroportuaria es un sistema diseñado para facilitar la administración y operación de un aeropuerto. Este programa tiene como objetivo mejorar la eficiencia en la gestión de vuelos y pasajeros mediante una interfaz sencilla y funcionalidades clave. A continuación, se detallan las funcionalidades ya integradas en el sistema:

Funcionalidades Integradas:

1. Permite registrar nuevos vuelos en el sistema, incluyendo información como el número de vuelo, origen, destino, horarios de salida y llegada.
2. Facilita la asignación de pasajeros a vuelos específicos. Se pueden ingresar los datos del pasajero y vincularlos a un vuelo particular.
3. Proporciona una lista completa de todos los vuelos registrados en el sistema, permitiendo visualizar detalles importantes de cada uno.
4. Muestra la lista de pasajeros asociados a un vuelo específico, lo cual es útil para el control de embarque y gestión de pasajeros.

Objetivos del Proyecto:

·Escalabilidad: Crear una base sólida para futuras expansiones del sistema, permitiendo la integración de nuevas funcionalidades según las necesidades del aeropuerto.

Este programa busca modernizar la gestión aeroportuaria, haciendo que las operaciones sean más fluidas y efectivas en todos los aspectos relacionados con vuelos y pasajeros.

Pruebas de Caja Negra

Las pruebas de caja negra se centran en la funcionalidad del sistema sin considerar su estructura interna. Aquí se prueban las entradas y salidas del sistema.

Prueba 1: Registro de nuevos vuelos

- Objetivo: Verificar que el sistema permite registrar un nuevo vuelo correctamente.
- Entrada:
- Origen: "Madrid"
- Destino: "Barcelona"

Procedimiento:

1. Ingresar los datos del vuelo.

- Resultado esperado: El vuelo se registra correctamente y aparece en la lista de vuelos.

```
1. Agregar un vuelo
2. Agregar un pasajero a un vuelo
3. Mostrar todos los vuelos
4. Mostrar todos los pasajeros de un vuelo
0. Terminar
1
Agregar un vuelo
Ingrese el origen del vuelo:
Madrid
Ingrese el destino del vuelo:
Barcelona
Vuelo agregado exitosamente
Selecciona una opcion:
1. Agregar un vuelo
2. Agregar un pasajero a un vuelo
3. Mostrar todos los vuelos
4. Mostrar todos los pasajeros de un vuelo
0. Terminar
3
Mostrar todos los vuelos
Origen: Madrid, Destino: Barcelona
```

Prueba 2: Asignación de pasajeros a vuelos

- Objetivo: Verificar que el sistema permite asignar un pasajero a un vuelo
- Entrada:
- Nombre: "Tu"
- Nombre: "El"
- Origen: "Madrid"
- Destino: "Barcelona"

Procedimiento:

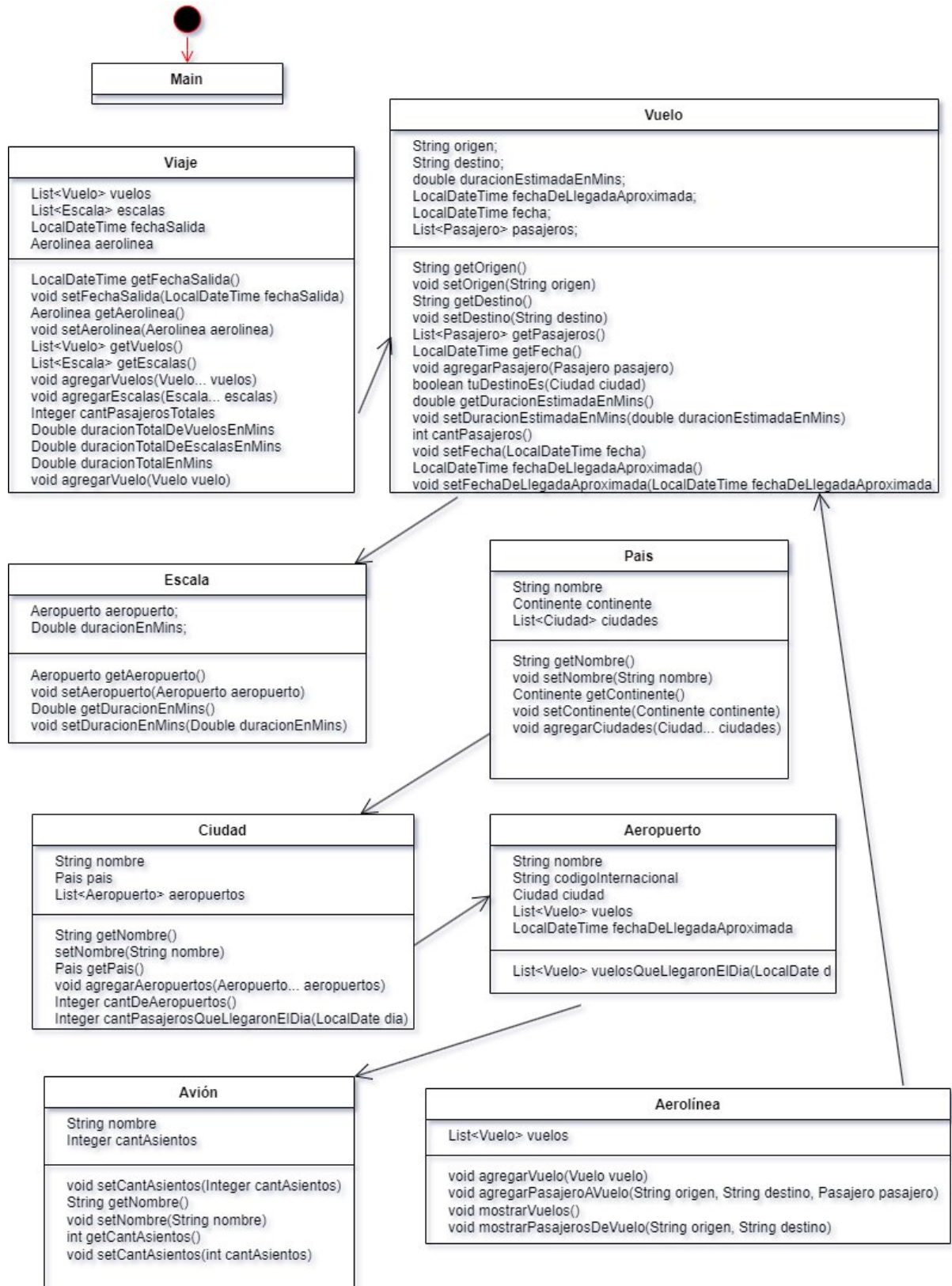
1. Ingresar los datos del pasajero.

2. Asignar el vuelo a ese pasajero.

- Resultado esperado: El pasajero se asigna correctamente al vuelo y aparece en la lista de pasajeros del vuelo.

```
2
Agregar un pasajero a un vuelo
Ingrese el nombre del pasajero:
Tu
Ingrese el apellido del pasajero:
El
Ingrese el origen del vuelo:
Madrid
Ingrese el destino del vuelo:
Barcelona
Pasajero agregado exitosamente
Selecciona una opcion:
1. Agregar un vuelo
2. Agregar un pasajero a un vuelo
3. Mostrar todos los vuelos
4. Mostrar todos los pasajeros de un vuelo
0. Terminar
4
Mostrar todos los pasajeros de un vuelo
Ingrese el origen del vuelo:
Madrid
Ingrese el destino del vuelo:
Barcelona
Tu El
```

Esquema UML



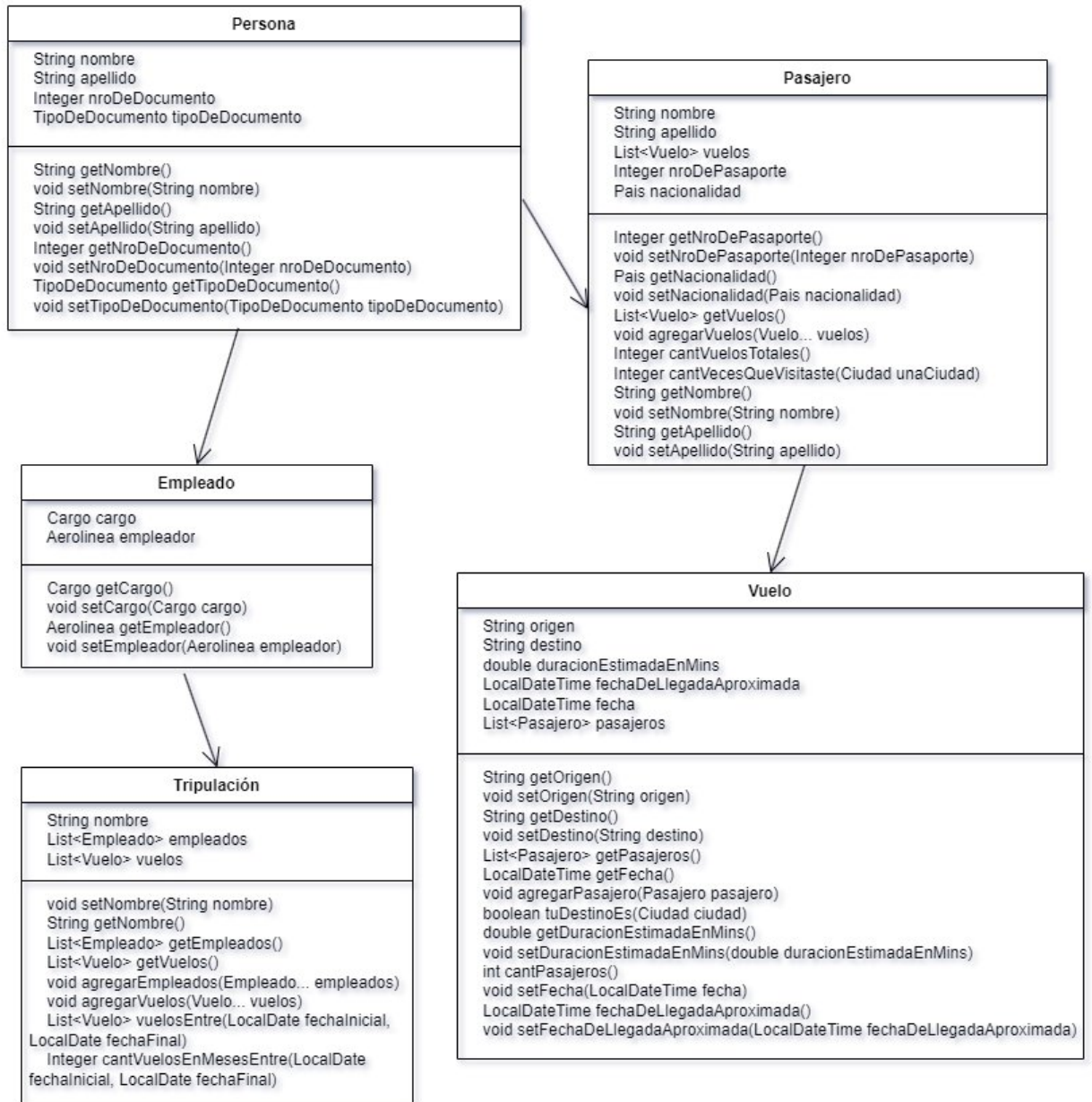
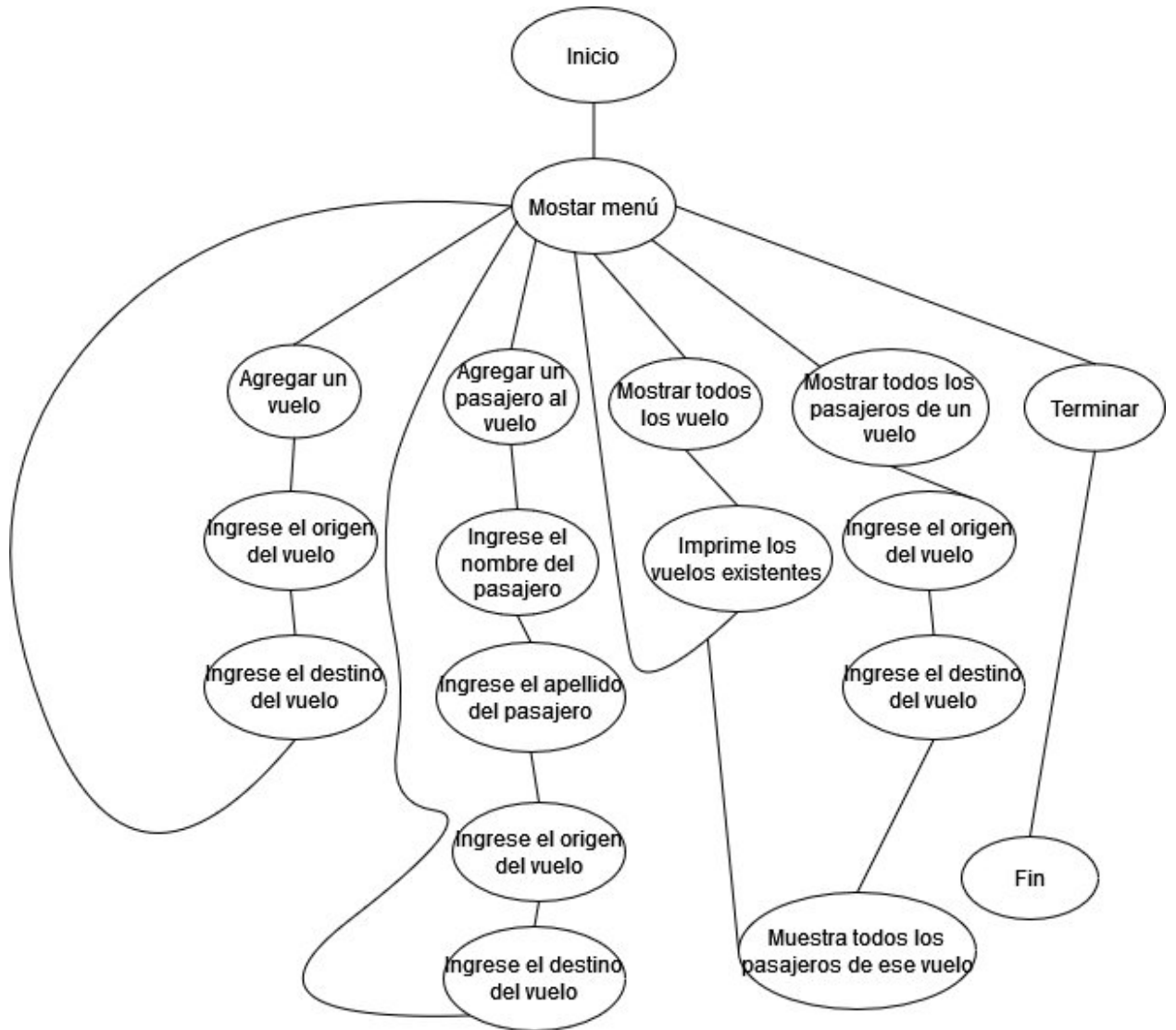


Diagrama de flujo



JavaDoc

PACKAGE CLASS USE TREE INDEX HELP

SEARCH

All Classes and Interfaces

All Classes and Interfaces	Classes	Enum Classes
Class	Description	
Aerolinea	La clase Aerolinea representa una aerolínea que gestiona una lista de vuelos.	
Aeropuerto	La clase Aeropuerto representa un aeropuerto con sus vuelos y su ciudad.	
Avion	La clase Avion representa un avión con su nombre y cantidad de asientos.	
Cargo	El enum Cargo representa los diferentes cargos disponibles en una aerolínea.	
Ciudad	La clase Ciudad representa una ciudad que contiene aeropuertos.	
Continente	El enum Continente representa los diferentes continentes del mundo.	
Empleado	La clase Empleado representa a un individuo empleado en una aerolínea, extendiendo la clase Persona.	
Escala	La clase Escala representa una escala en un vuelo, con un aeropuerto de escala y una duración en minutos.	
Main	La clase principal para ejecutar el sistema de gestión de aerolíneas.	
Pais	La clase Pais representa un país con un nombre, un continente al que pertenece y una lista de ciudades.	
Pasajero	La clase Pasajero representa a una persona que viaja en vuelos, con información como nombre, apellido, número de pasaporte, nacionalidad y una lista de vuelos.	
Persona	La clase abstracta Persona representa a una persona con información básica como nombre, apellido, número de documento y tipo de documento.	
TipoDeDocumento	El enum TipoDeDocumento define los diferentes tipos de documentos de identidad.	
Tripulacion	La clase Tripulacion representa a un grupo de empleados que trabajan juntos en varios vuelos.	
Viaje	La clase Viaje representa un conjunto de vuelos y escalas que forman parte de un viaje, con una fecha de salida y una aerolínea.	
Vuelo	La clase Vuelo representa un vuelo entre un origen y un destino, con una duración estimada, fechas y una lista de pasajeros.	

Todo el javadoc y sus correspondientes clases creados automáticamente por IntelliJ Idea se encuentran en el siguiente enlace con sus correspondientes argumentos y funcionalidades.

[Aqui el JavaDoc](#)

Funciones futuras y modularización

Se crean muchas clases distintas con muchos atributos y funcionalidades.

Muchas de estas se han erradicado al final del proyecto.

Otras en cambio permanecen para generar un proyecto modular y extensible de cara a futuro. Se podrán añadir tales funciones como distintas aerolíneas, datos adicionales de los pasajeros tales como pasaportes.

Aparte de esto en algún momento se añadirá funcionalidad gráfica.