

Manual de usuario

Harold Guerrero Caicedo

Juan Esteban Salas

Proyecto #1

Programación Orientada a Objetos

Introducción

¡Bienvenido al Sistema de Gestión de Aeronaves y Vuelos! Este sistema está diseñado para ayudarte a administrar aeronaves, vuelos, tripulaciones y pasajeros. A través de esta guía, aprenderás cómo utilizar las diversas funcionalidades del sistema.

Tabla de Contenidos

1. Inicio de Sesión

- Ingreso como administrador.

2. Menú Principal

- 2.1 Comprar un vuelo.
- 2.2 Consultar puertas de embarque.
- 2.3 Consultar vuelos.
- 2.4 Ver opciones de administrador.

3. Opciones del Administrador

- 3.1 Agregar una nave.
- 3.2 Consultar información de las naves.
- 3.3 Editar la información de una nave.
- 3.4 Generar un vuelo.
- 3.5 Ingresar una tripulación al sistema.
- 3.6 Consultar tripulaciones.
- 3.7 Consultar historial de una puerta de embarque.
- 3.8 Despachar un vuelo.

4. Anexo: Documentación del código

1. Inicio de Sesión

Para acceder al sistema como administrador, se requiere una contraseña. Deberás ingresar la contraseña de administrador al iniciar sesión. Tendrás tres intentos para ingresar la contraseña correctamente. Si fallas tres veces, el sistema se bloqueará por seguridad y deberás comunicarte con el administrador.

2. Menú Principal

Después de iniciar sesión, accederás al menú principal, que ofrece diversas opciones:

2.1 Comprar un Vuelo

Esta opción te permite comprar un vuelo ingresando tus datos como pasajero y seleccionando un vuelo disponible. También puedes consultar los vuelos antes de tomar una decisión.

2.2 Consultar Puertas de Embarque

Aquí podrás ver la lista de puertas de embarque disponibles y su información asociada.

2.3 Consultar Vuelos

Puedes explorar la lista de vuelos disponibles y, si eres administrador, acceder a información adicional sobre pasajeros y tripulación.

2.4 Ver Opciones de Administrador

Si eres administrador, podrás acceder a un conjunto de opciones adicionales para gestionar el sistema.

3. Opciones del Administrador

Si seleccionas "Ver Opciones de Administrador" en el menú principal, podrás acceder a las siguientes funcionalidades:

3.1 Agregar una Nave

Puedes registrar una nueva aeronave ingresando sus detalles, como registro, capacidad, velocidad máxima, autonomía, año de fabricación y tipo (avión, helicóptero o jet privado).

3.2 Consultar Información de las Naves

Podrás ver detalles de las aeronaves registradas en el sistema.

3.3 Editar la Información de una Nave

Modificar características de una aeronave, como registro, capacidad, velocidad, autonomía, año de fabricación, estado y atributos específicos según el tipo.

3.4 Generar un Vuelo

Crear un nuevo vuelo, asignando la ciudad de origen y destino, fecha, aeronave, puerta de embarque y tripulación.

3.5 Ingresar una Tripulación al Sistema

Registrar datos de la tripulación, incluyendo nacionalidad, número de maletas, información médica, cédula, nombres, apellidos, fecha de nacimiento, género, dirección, número de teléfono, correo, puesto, años de experiencia y cantidad máxima de horas diarias de trabajo.

3.6 Consultar Tripulaciones

Ver información detallada de las tripulaciones registradas en el sistema.

3.7 Consultar Historial de una Puerta de Embarque

Obtener información sobre el historial de uso de una puerta de embarque específica.

3.8 Despachar un Vuelo

Marcar un vuelo como despachado, liberando la puerta de embarque y permitiendo su salida.

4. Documentación del código

Clase Aeronave

Esta es la clase base que representa una aeronave genérica.

Atributos:

- **registro_de_marca** (número de registro de la aeronave)
- **capacidad_de_pasajeros** (número de pasajeros que puede transportar)
- **velocidad_max** (velocidad máxima de la aeronave)
- **autonomía** (autonomía de vuelo en millas)
- **año_fabricación** (año de fabricación de la aeronave)
- **estado** (estado actual de la aeronave, por ejemplo, "en el vuelo" o "en tierra")
- **coordenadas** (un par de coordenadas de latitud y longitud)
- **coor_aeronaves** (un vector que almacena las coordenadas de todas las aeronaves)
- **id** (identificador para la torre de control)

Funciones:

- **guardar_coordenadas**: genera coordenadas aleatorias y las asigna a la aeronave.
- **obtenerCoordenadas**: devuelve las coordenadas de la aeronave.
- **consultarCoordenadasAero**: permite actualizar las coordenadas de una aeronave específica en el vector "coor_aeronaves".
- **reiniciar**: actualiza las coordenadas de todas las aeronaves en función de un vector de coordenadas proporcionado.
- **getTipo**: Retorna el tipo de aeronave entre las subclases Avion, Helicoptero y JetPrivado.
- **ImprimirInformacionEspecificas**: Función virtual que imprime la información específica de las clases derivadas de Aeronave. Se sobrescribe para cada subclase.

Clase Avión (hereda de la clase Aeronave)

Representa un tipo de aeronave que es un avión.

Atributos adicionales:

- **altitud_maxima** (la latitud máxima a la que puede volar)
- **cantidad_de_motores** (número de motores del avión)
- **categoría** (categoría de avión)

El constructor de esta clase llama al constructor de la clase base "Aeronave" y establece atributos específicos del Avión.

Clase Helicóptero (hereda de la clase Aeronave)

Representa un tipo de aeronave que es un helicóptero.

Atributos adicionales:

- ***cantidad_de_rotor*** (número de rotores del helicóptero)
- ***capacidad_de_elevación*** (capacidad de carga que puede elevar)
- ***uso_especifico*** (propósito o uso específico del helicóptero)

El constructor de esta clase llama al constructor de la clase base "Aeronave" y establece atributos específicos del helicóptero.

Clase Jetprivado (hereda de la clase Aeronave)

Representa un tipo de aeronave que es un jet privado.

Atributos adicionales:

- ***propietario*** (nombre del propietario del jet privado)
- ***lista_de_servicios_a_bordo*** (lista de servicios disponibles a bordo)
- ***lista_de_destinos_frecuentes*** (lista de destinos frecuentes para esta aeronave)

El constructor de esta clase llama al constructor de la clase base "Aeronave" y establece los atributos específicos del jet privado.

Clase PuertaDeEmbarque

Esta clase representa una puerta de embarque. Se utiliza para gestionar y registrar información relacionada con las puertas de embarque de un aeropuerto, incluyendo detalles sobre el vuelo asignado, su disponibilidad y el historial de vuelo que han utilizado esa puerta en el pasado.

Atributos:

- ***identificacion*** (identificador de la puerta de embarque)
- ***ubicacion*** (ubicación de la puerta de embarque)
- ***vuelo_asignado*** (puntero al vuelo asignado a la puerta de embarque)
- ***disponible*** (indica si la puerta de embarque está disponible)
- ***hora_embarque*** (hora de embarque)

- **historial** (vector que almacena el historial de vuelos asignados a la puerta de embarque)

Funciones:

- **Constructor:** recibe un identificador y una ubicación para crear una puerta de embarque.
- **Getters y Setters** para acceder y modificar los atributos de la puerta de embarque.
- **getVueloAsignado:** devuelve el vuelo asignado a la puerta de embarque.
- **setVueloAsignado:** asigna un vuelo a la puerta de embarque.
- **getDisponible:** devuelve si la puerta de embarque está disponible.
- **setDisponible:** establece si la puerta de embarque está disponible.
- **getHistorial:** devuelve el historial de vuelos asignados a la puerta de embarque.
- **agregarHistorial:** agrega un vuelo al historial de la puerta de embarque.
- **imprimirInfo:** función auxiliar para imprimir información sobre la puerta de embarque.
- **imprimirHistorial:** función auxiliar para imprimir el historial de vuelos asignados a la puerta de embarque.

Clase TorreDeControl

La clase TorreDeControl representa una entidad fundamental en la gestión y coordinación de un aeropuerto. Su propósito principal es administrar y mantener actualizada la información relacionada con las aeronaves, las puertas de embarque y los vuelos. A continuación, se detalla la estructura y las funciones de esta clase.

Descripción:

La Torre de Control se encarga de entregar y recibir información sobre las aeronaves. Se hace énfasis en mantener las coordenadas de las aeronaves actualizadas, lo que permite a todas las aeronaves conocer la ubicación en "tiempo real" de las demás. Además, también se encarga de asignar y manipular las puertas de embarque.

Atributos:

- **coordenadas_aeronaves:** Un vector que almacena las coordenadas de todas las aeronaves registradas en el sistema. Esto permite un seguimiento en tiempo real de la ubicación de cada aeronave.
- **puertas_embarques:** Un vector que representa las puertas de embarque disponibles en el aeropuerto. Cada puerta se almacena como una instancia de la clase PuertaDeEmbarque.
- **vuelos:** Un vector de punteros a objetos de la clase Vuelo, que mantiene información sobre los vuelos programados y en curso.

- **aero:** Un vector de punteros únicos (unique_ptr) a objetos de la clase base Aeronave. Esta estructura permite almacenar punteros a cualquier tipo de clase derivada de Aeronave, como Avión, Helicóptero o JetPrivado.

Funciones:

- **Constructor:** La Torre de Control se inicia con un constructor que recibe el número de puertas de embarque disponibles. Desde la creación, se definen y crean las puertas de embarque necesarias en el aeropuerto.
- **actualizar_ubi_aero:** Actualiza la ubicación de una sola aeronave y la guarda en el registro de coordenadas. Esta función es esencial para mantener la información de ubicación actualizada.
- **obtener_ubi:** Devuelve las coordenadas de una aeronave específica identificada por su ID. Esto facilita a otras aeronaves conocer la ubicación de una aeronave en particular.
- **agregarAeronave:** Agrega una aeronave al sistema utilizando un puntero único (unique_ptr). Esta función permite gestionar y almacenar diferentes tipos de aeronaves de manera eficiente.
- **actualizarTodasLasAeronaves:** Actualiza toda la información de coordenadas para todas las aeronaves registradas en el sistema. Esto garantiza que todas las aeronaves tengan acceso a la información actualizada de ubicación de las demás.
- **asignar_puertas:** Asigna una puerta de embarque a un vuelo específico. Esta función es útil al generar un vuelo y garantiza que cada vuelo tenga una puerta de embarque asignada.
- **desocupar_puerta:** Libera una puerta de embarque previamente asignada. Esto se utiliza cuando un vuelo ha partido o ha finalizado su proceso de embarque.
- **hayPuertasDisponibles:** Verifica si hay puertas de embarque disponibles y devuelve el número de la puerta disponible. Esta función es útil al asignar una puerta a un vuelo.
- **hayAeronavesDisponibles:** Verifica si hay aeronaves disponibles para asignar a un vuelo y devuelve el número de la aeronave disponible. Esto es útil al programar un nuevo vuelo.

La clase TorreDeControl desempeña un papel central en la operación y gestión de un aeropuerto, permitiendo una coordinación eficiente entre las aeronaves, las puertas de embarque y los vuelos programados.

Clase Pasajero

La clase Pasajero representa a un individuo que viaja en una aeronave y que tiene información personal relevante para el proceso de abordaje y seguridad. Esta clase almacena detalles como nacionalidad, cantidad de maletas, información médica, identificación, nombres, apellidos, fecha de nacimiento, género, dirección, número de

teléfono y correo electrónico. Además, se incluye un campo para registrar el número de vuelo al que está asociado el pasajero.

Atributos:

- ***nacionalidad:*** Almacena la nacionalidad del pasajero.
- ***numero_maletas:*** Indica la cantidad de maletas que el pasajero planea llevar consigo en el vuelo.
- ***resumen_infomedica:*** Contiene información médica relevante sobre el pasajero.
- ***cedula:*** Almacena el número de cédula de identidad del pasajero.
- ***nombres:*** Guarda los nombres del pasajero.
- ***apellidos:*** Almacena los apellidos del pasajero.
- ***fecha_de_nacimiento:*** Indica la fecha de nacimiento del pasajero.
- ***genero:*** Registra el género del pasajero.
- ***direccion:*** Almacena la dirección de residencia del pasajero.
- ***numero_telefono:*** Guarda el número de teléfono del pasajero.
- ***correo:*** Almacena la dirección de correo electrónico del pasajero.
- ***NoVuelo:*** Guarda el número de vuelo al que está asociado el pasajero.

Funciones:

- ***setVuelo:*** Permite asignar el número de vuelo al pasajero.
- ***obtener_numero_maletas:*** Obtiene la cantidad de maletas que el pasajero planea llevar en el vuelo.
- ***poner_numero_maletas:*** Permite establecer la cantidad de maletas que el pasajero llevará en el vuelo.
- ***imprimirInfo:*** Imprime la información relevante del pasajero.

Clase Tripulación (hereda de Pasajero)

La clase Tripulación es una clase derivada de la clase Pasajero. Representa a los miembros de la tripulación de una aeronave, como pilotos o asistentes de vuelo. Además de los atributos heredados de Pasajero, esta clase incluye detalles específicos de la tripulación, como el puesto, la cantidad de años de experiencia y la cantidad máxima de horas de trabajo diarias.

Atributos adicionales:

- ***puesto:*** Indica el puesto o función que desempeña el miembro de la tripulación (por ejemplo, piloto o asistente de vuelo).

- ***anios_experiencia:*** Almacena la cantidad de años de experiencia en el trabajo de la aviación.
- ***cant_max_horasdiarias:*** Indica la cantidad máxima de horas de trabajo diarias permitidas para el miembro de la tripulación.

Funciones adicionales:

- ***imprimirInfoTripulacion:*** Imprime la información específica de la tripulación, incluyendo el puesto, los años de experiencia y la cantidad máxima de horas de trabajo diarias.

La clase Pasajero y su clase derivada, Tripulación, son esenciales en la gestión de pasajeros y personal de vuelo en una aeronave, permitiendo registrar y organizar la información relevante para garantizar un viaje seguro y cómodo.

Clase Vuelo

La clase Vuelo representa un vuelo específico que parte desde una ciudad de origen hacia una ciudad de destino en una fecha determinada. Esta clase almacena información relevante sobre el vuelo, incluyendo detalles sobre la tripulación y los pasajeros que estarán a bordo.

Atributos:

- ***tripulacion:*** Un vector que contiene punteros a objetos de la clase Tripulacion, representando a los miembros de la tripulación que estarán a cargo del vuelo.
- ***pasajeros:*** Un vector que contiene punteros a objetos de la clase Pasajero, representando a los pasajeros que estarán a bordo del vuelo.
- ***numero_de_identificacion:*** Un número de identificación único para el vuelo.
- ***fecha:*** La fecha en la que el vuelo está programado para partir.
- ***ciudad_origen:*** La ciudad de origen desde la cual partirá el vuelo.
- ***ciudad_destino:*** La ciudad de destino a la cual se dirigirá el vuelo.
- ***puerta_embarque:*** El número de la puerta de embarque asignada para el vuelo.
- ***idAeronave:*** Un identificador único de la aeronave que realizará el vuelo.

Constructores:

- ***Vuelo():*** Constructor por defecto.
- ***Vuelo(Vuelo* vueloPtr):*** Constructor que crea una copia de otro objeto Vuelo.
- ***Vuelo(string num_identificacion, string fecha_vuelo, string origen, string destino, int puerta):*** Constructor que inicializa los atributos del vuelo con valores proporcionados.

Funciones:

- ***agregarTripulante:*** Agrega un miembro de la tripulación al vuelo mediante un puntero a un objeto de la clase Tripulación.
- ***AgregarPasajero:*** Agrega un pasajero al vuelo mediante un puntero a un objeto de la clase Pasajero.

La clase Vuelo es fundamental para gestionar y organizar la información relacionada con un vuelo específico, incluyendo los miembros de la tripulación y los pasajeros asignados a ese vuelo. Esto permite un seguimiento preciso de quiénes estarán a bordo y quiénes estarán a cargo de la operación del vuelo.