Imagen que contiene Texto

Descripción generada automáticamente

Tabla de contenido

[Descripción General 3](#_Toc175520485)

[Clases y Métodos 3](#_Toc175520486)

[ Clase VehiculoAereo 3](#_Toc175520487)

[o Atributos Privados: 3](#_Toc175520488)

**Control de vuelos de un aeropuerto.**

# Descripción General

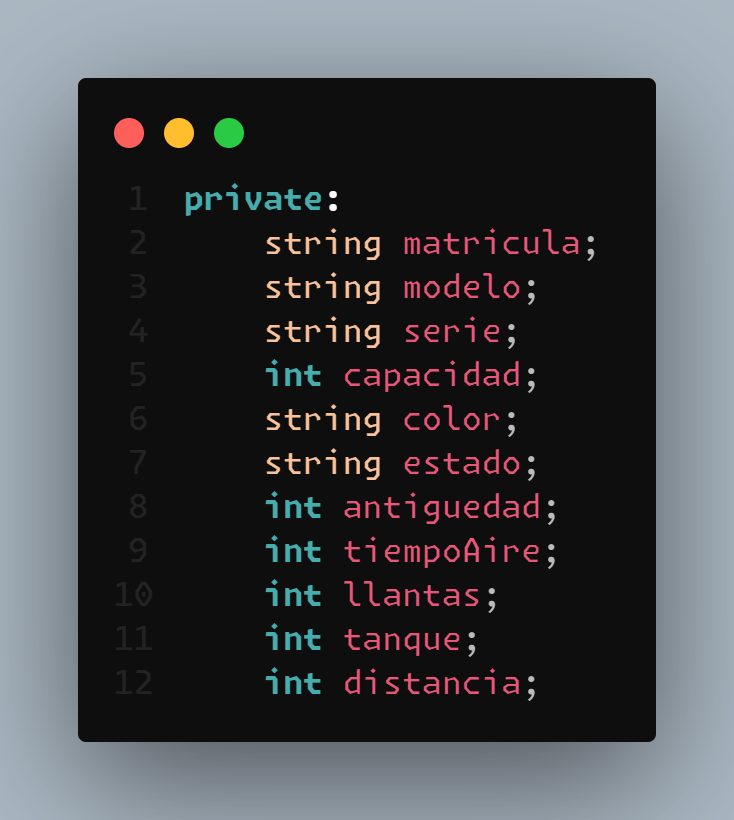
Este programa en C++ gestiona información sobre vehículos aéreos, tripulación, pasajeros y vuelos. Incluye clases como VehiculoAereo, Avion, Avioneta, Persona, Tripulacion, Pasajero, Vuelo, entre otras. Además, permite almacenar y recuperar datos en formato JSON.

# Clases y Métodos

## Clase VehiculoAereo

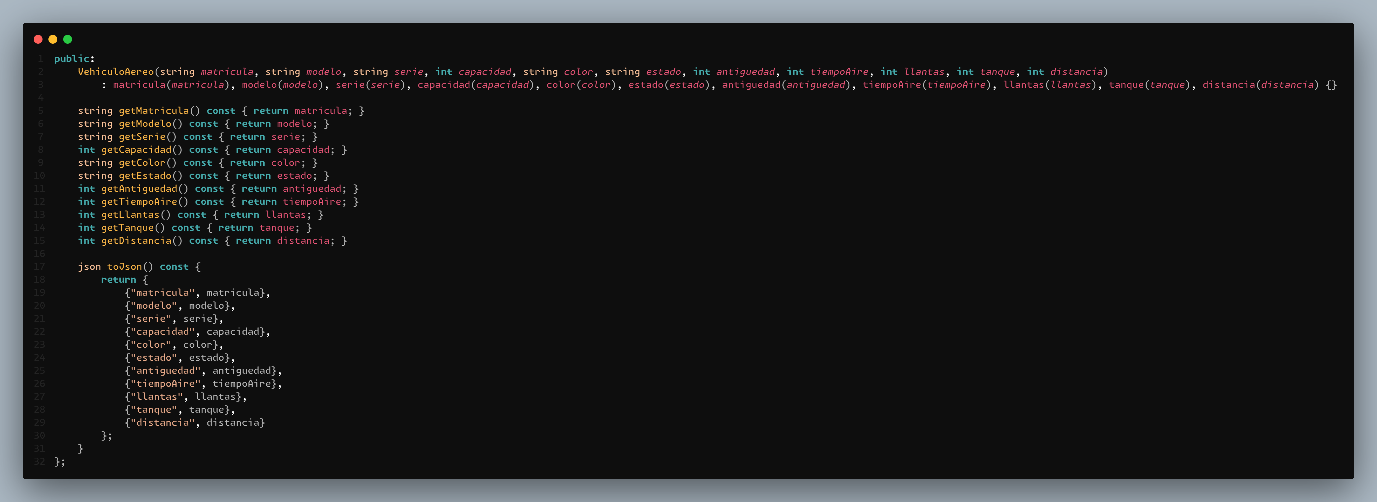
## Atributos Privados:

* + - string matricula: Matrícula del vehículo.
    - string modelo: Modelo del vehículo.
    - string serie: Serie del vehículo.
    - int capacidad: Capacidad del vehículo.
    - string color: Color del vehículo.
    - string estado: Estado actual del vehículo.
    - int antiguedad: Antigüedad del vehículo.
    - int tiempoAire: Tiempo en el aire del vehículo.
    - int llantas: Número de llantas del vehículo.
    - int tanque: Capacidad del tanque del vehículo.
    - int distancia: Distancia máxima que puede recorrer el vehículo.



### Métodos Públicos:

* + - **Constructor:** Inicializa todos los atributos.
    - **Getters:** Métodos para obtener cada atributo.
    - **json toJson()**: Convierte los datos de la clase a formato JSON.



## Clase Avion (hereda de VehiculoAereo)

### Atributos Privados:

* + - string aerolinea: Aerolínea propietaria del avión.
    - string tipoMotor: Tipo de motor del avión.
    - int puertas: Número de puertas del avión.
    - string tipoAvion: Tipo de avión.

Texto

Descripción generada automáticamente

### Métodos Públicos:

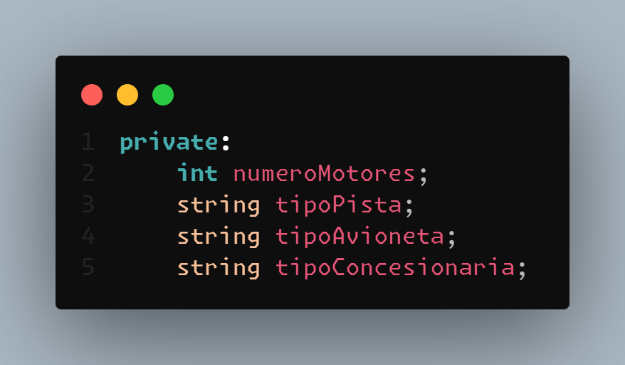
* + - **Constructor:** Inicializa todos los atributos, incluyendo los heredados de VehiculoAereo.
    - **Getters:** Métodos para obtener cada atributo.
    - **json toJson()**: Convierte los datos de la clase a formato JSON.



## Clase Avioneta (hereda de VehiculoAereo)

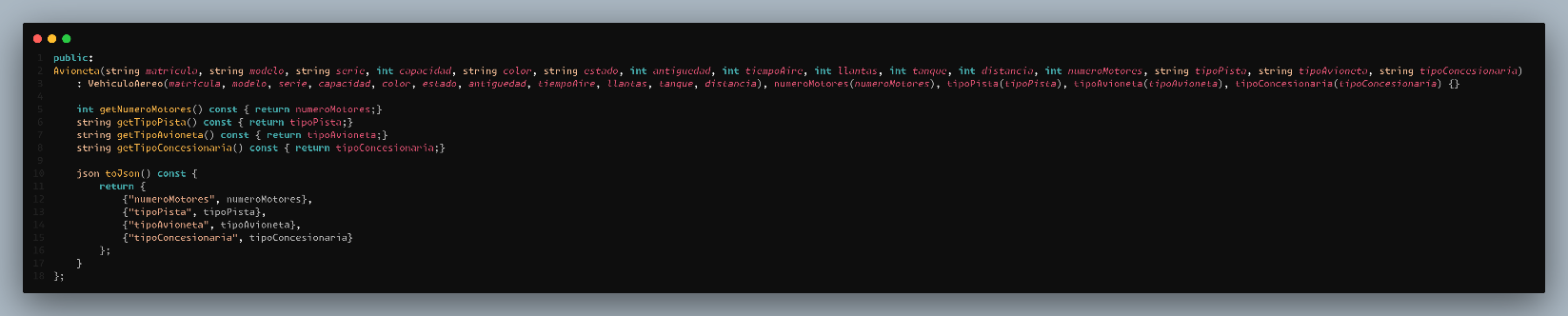
### Atributos Privados:

* + - int numeroMotores: Número de motores de la avioneta.
    - string tipoPista: Tipo de pista requerida.
    - string tipoAvioneta: Tipo de avioneta.
    - string tipoConcesionaria: Concesionaria de la avioneta.



### Métodos Públicos:

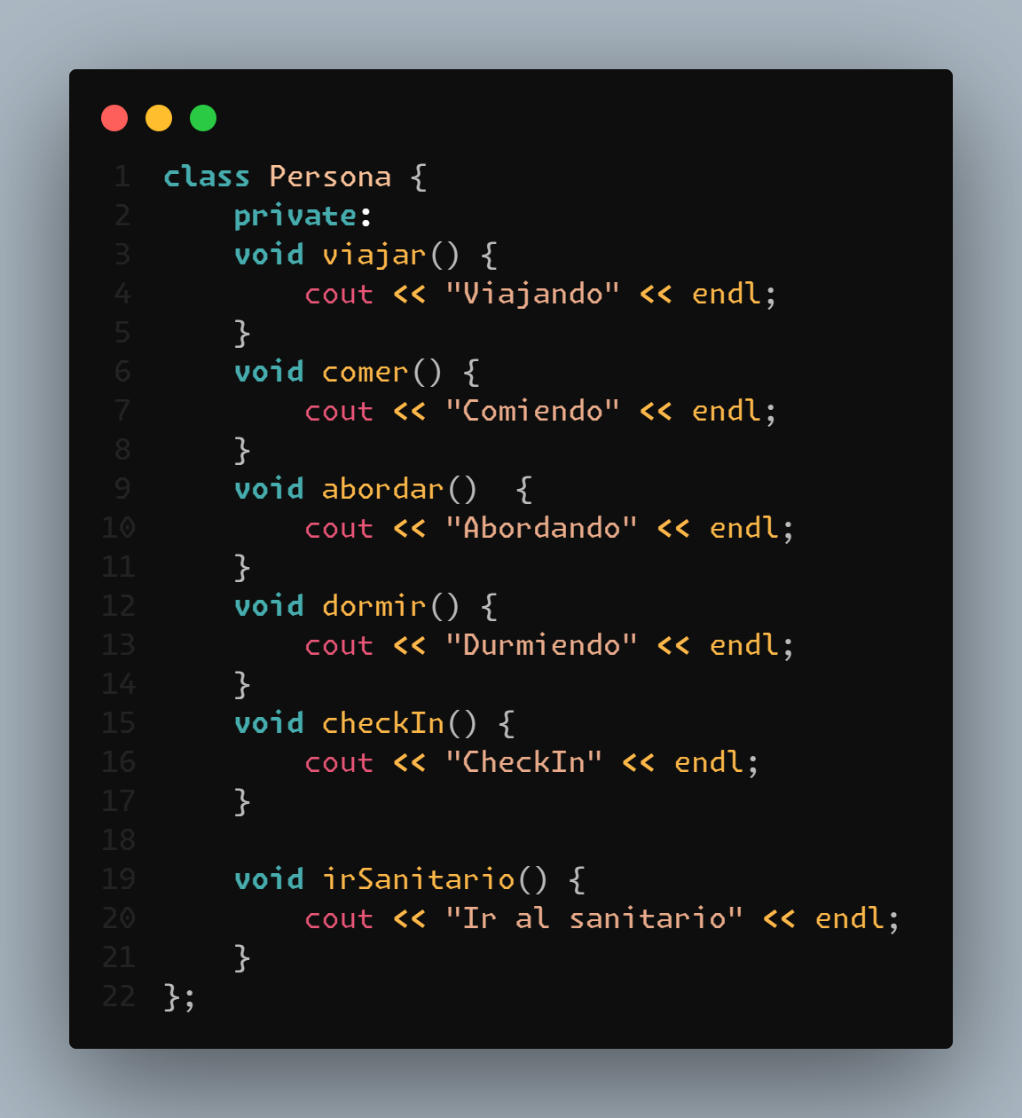
* + - **Constructor:** Inicializa todos los atributos, incluyendo los heredados de VehiculoAereo.
    - **Getters:** Métodos para obtener cada atributo.
    - **json toJson()**: Convierte los datos de la clase a formato JSON.



## Clase Persona

### Métodos Privados:

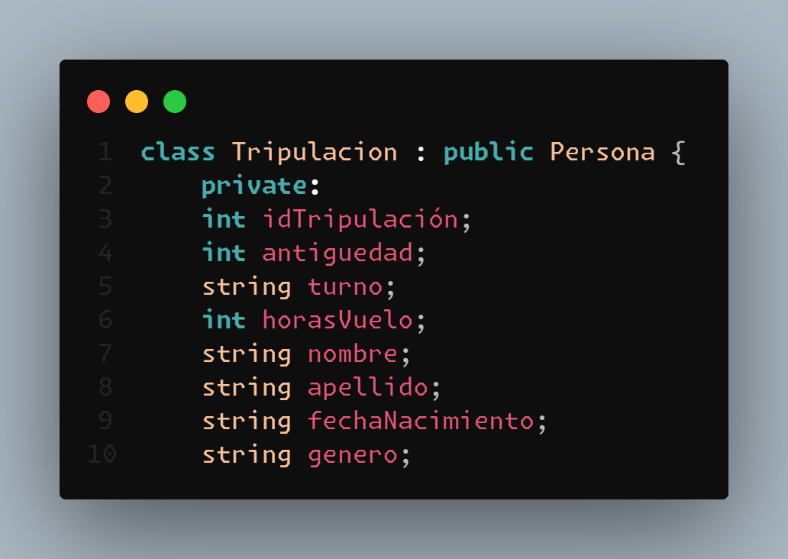
* + - void viajar(): Simula el acto de viajar.
    - void comer(): Simula el acto de comer.
    - void abordar(): Simula el acto de abordar.
    - void dormir(): Simula el acto de dormir.
    - void checkIn(): Simula el acto de hacer check-in.
    - void irSanitario(): Simula el acto de ir al sanitario.



## Clase Tripulacion (hereda de Persona)

### Atributos Privados:

* + - int idTripulacion: Identificador de la tripulación.
    - int antiguedad: Antigüedad del miembro de la tripulación.
    - string turno: Turno de trabajo.
    - int horasVuelo: Horas de vuelo acumuladas.
    - string nombre: Nombre del miembro de la tripulación.
    - string apellido: Apellido del miembro de la tripulación.
    - string fechaNacimiento: Fecha de nacimiento del miembro de la tripulación.
    - string genero: Género del miembro de la tripulación.



### Métodos Públicos:

* + - **Constructor:** Inicializa todos los atributos.
    - **Getters:** Métodos para obtener cada atributo.
    - **json toJson()**: Convierte los datos de la clase a formato JSON.



## Clase Piloto (hereda de Tripulacion)

### Atributos Privados:

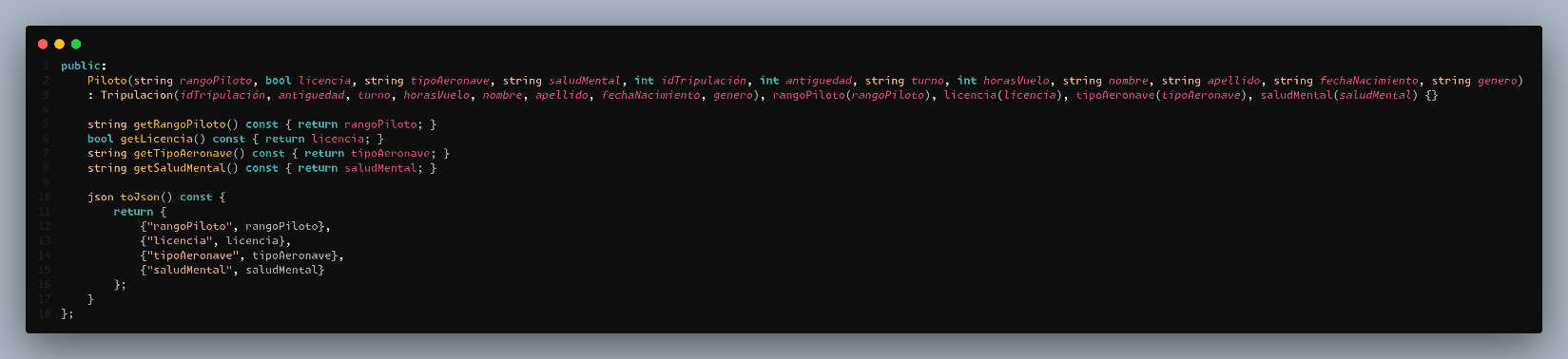
* + - string rangoPiloto: Rango del piloto.
    - bool licencia: Estado de la licencia del piloto.
    - string tipoAeronave: Tipo de aeronave que puede pilotar.
    - string saludMental: Estado de salud mental del piloto.

Texto

Descripción generada automáticamente

### Métodos Públicos:

* + - Constructor**:** Inicializa todos los atributos, incluyendo los heredados de Tripulacion.
    - Getters**:** Métodos para obtener cada atributo.
    - jsontoJson**()**: Convierte los datos de la clase a formato JSON.



## Clase Copiloto (hereda de Tripulacion)

### Atributos Privados:

* + - string rangoCopiloto: Rango del copiloto.
    - bool licencia: Estado de la licencia del copiloto.
    - string idioma: Idioma que habla el copiloto.
    - int certificados: Número de certificados del copiloto.

Texto

Descripción generada automáticamente

### Métodos Públicos:

* + - Constructor**:** Inicializa todos los atributos, incluyendo los heredados de Tripulacion.
    - Getters**:** Métodos para obtener cada atributo.
    - jsontoJson**()**: Convierte los datos de la clase a formato JSON.



## Clase Sobrecargo (hereda de Tripulacion)

#### Atributos Privados:

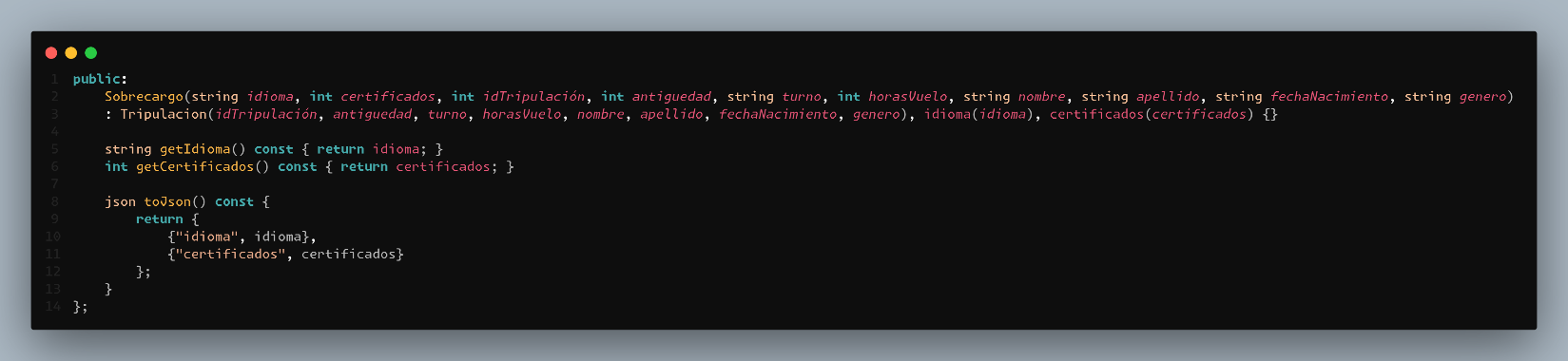
* + - string idioma: Idioma que habla el sobrecargo.
    - int certificados: Número de certificados del sobrecargo.

Texto

Descripción generada automáticamente

#### Métodos Públicos:

* + - Constructor**:** Inicializa todos los atributos, incluyendo los heredados de Tripulacion.
    - Getters**:** Métodos para obtener cada atributo.
    - jsontoJson**()**: Convierte los datos de la clase a formato JSON.



## Clase Aeropuerto

#### Atributos Privados:

* + - string claveAeropuerto: Código del aeropuerto.
    - string nombre: Nombre del aeropuerto.
    - int cantidadPistas: Número de pistas en el aeropuerto.
    - string tipoAvion: Tipo de aviones que acepta el aeropuerto.



#### Métodos Públicos:

* + - Constructor**:** Inicializa todos los atributos.
    - Getters**:** Métodos para obtener cada atributo.
    - jsontoJson**()**: Convierte los datos de la clase a formato JSON.



## Clase Terminal

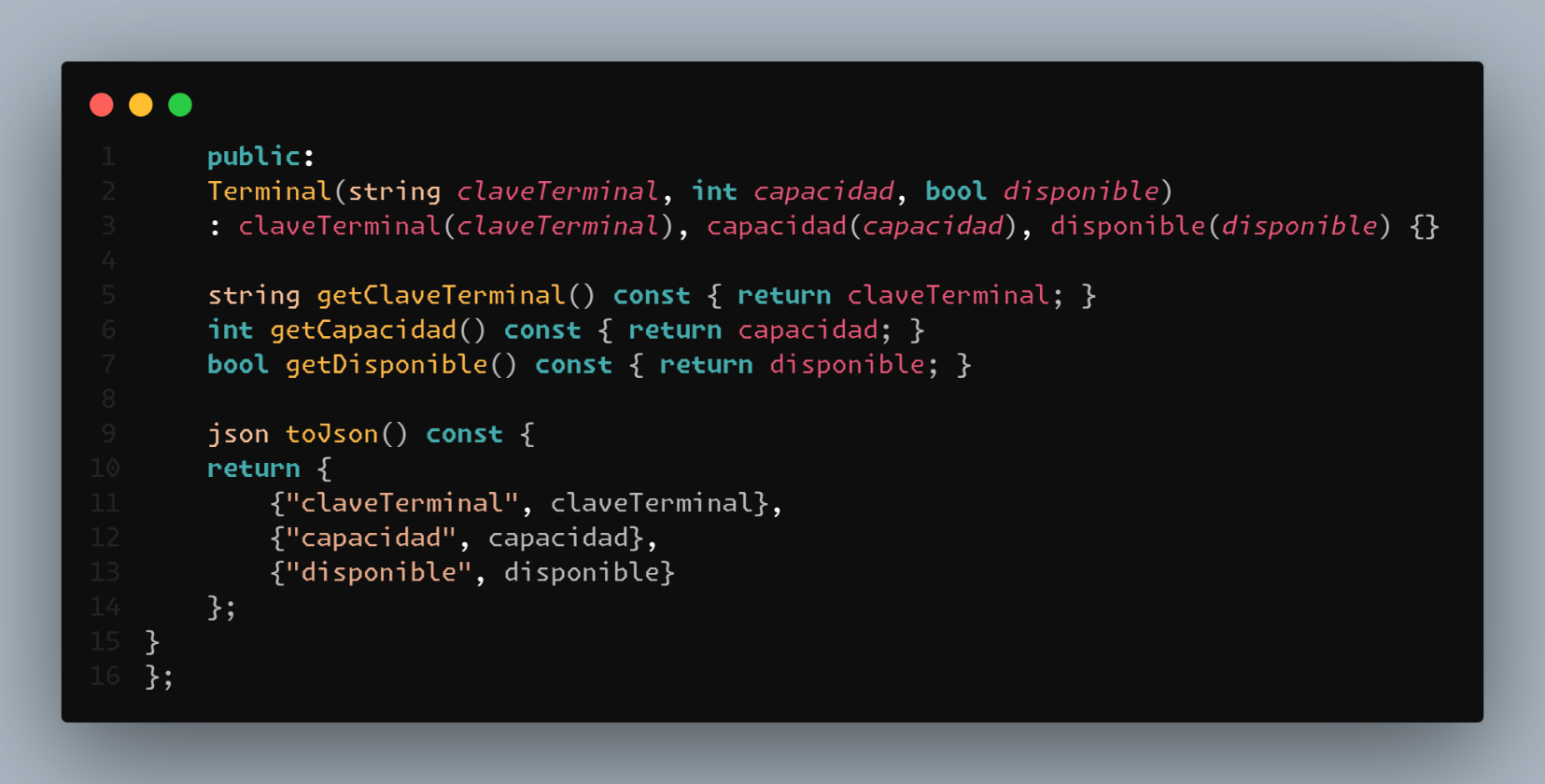
### Atributos Privados:

* + - string claveTerminal: Código de la terminal.
    - int capacidad: Capacidad de la terminal.
    - bool disponible: Disponibilidad de la terminal.



### Métodos Públicos:

* + - Constructor**:** Inicializa todos los atributos.
    - Getters**:** Métodos para obtener cada atributo.
    - jsontoJson**()**: Convierte los datos de la clase a formato JSON.



## Clase Equipaje

### Atributos Privados:

* + - double peso: Peso del equipaje.
    - double altura: Altura del equipaje.
    - double ancho: Ancho del equipaje.
    - int idPasajero: Identificador del pasajero propietario del equipaje.
    - string tipoEquipaje: Tipo de equipaje.

Pantalla de computadora con letras

Descripción generada automáticamente

### Métodos Públicos:

* + - Constructor**:** Inicializa todos los atributos.
    - Getters**:** Métodos para obtener cada atributo.
    - jsontoJson**()**: Convierte los datos de la clase a formato JSON.



## Clase Pasajero (hereda de Persona)

#### Atributos Privados:

* + - int id: Identificador del pasajero.
    - string nombre: Nombre del pasajero.
    - string apellido: Apellido del pasajero.
    - int edad: Edad del pasajero.
    - string direccion: Dirección del pasajero.
    - string fechaNacimiento: Fecha de nacimiento del pasajero.
    - bool discapacidad: Indica si el pasajero tiene alguna discapacidad.
    - string nacionalidad: Nacionalidad del pasajero.
    - Equipaje equipaje: Equipaje asociado al pasajero.

Texto

Descripción generada automáticamente

### Métodos Públicos:

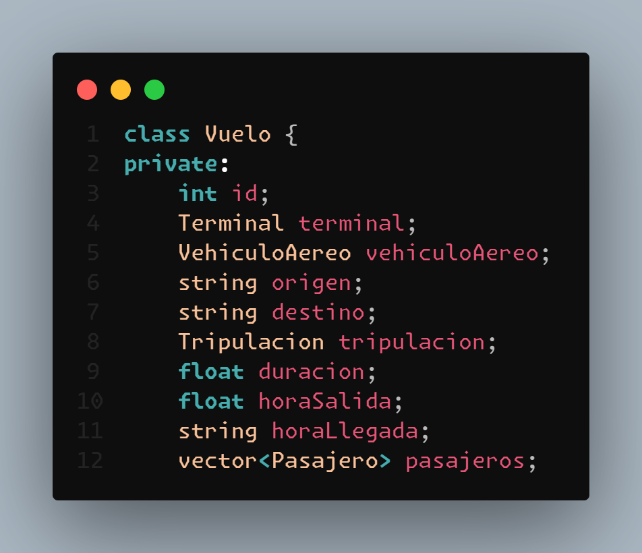
* + - Constructor**:** Inicializa todos los atributos.
    - Getters**:** Métodos para obtener cada atributo.
    - jsontoJson**()**: Convierte los datos de la clase a formato JSON.



## Clase Vuelo

### Atributos Privados:

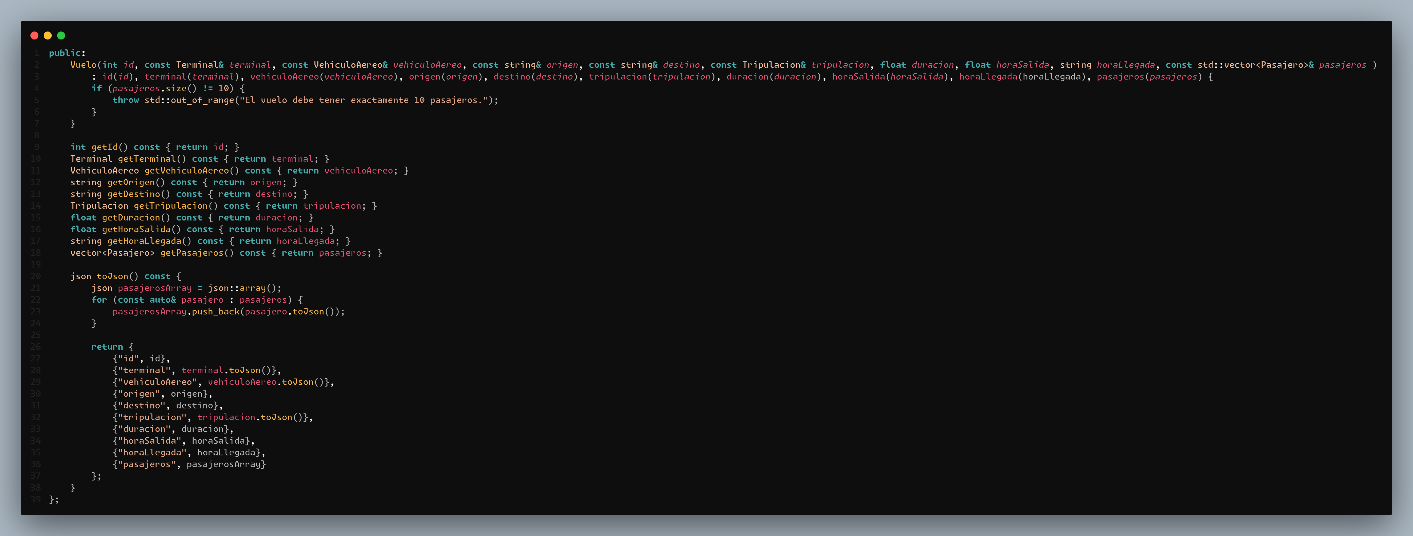
* + - int id: Identificador del vuelo.
    - Terminal terminal: Terminal de salida del vuelo.
    - VehiculoAereo vehiculoAereo: Vehículo aéreo asignado al vuelo.
    - string origen: Origen del vuelo.
    - string destino: Destino del vuelo.
    - Tripulacion tripulacion: Tripulación asignada al vuelo.
    - float duracion: Duración del vuelo.
    - float horaSalida: Hora de salida del vuelo.
    - string horaLlegada: Hora estimada de llegada del vuelo.
    - pasajeros: Vector que almacena los pasajeros asignados al vuelo (máximo de 10 pasajeros).
    - tripulacion: Vector que almacena los miembros de la tripulación asignados al vuelo.



### Métodos:

Constructor: Inicializa un objeto Vuelo con los valores proporcionados.

toJson**()**: Convierte los datos del vuelo en un formato JSON para facilitar su almacenamiento en un archivo.

****

Ejemplo:

Vuelo vuelo1(1, terminal1, avion1, "Ciudad de México", "Nueva York", tripulacion1, 5.5, 8.0, "13:30", pasajeros1);

Este constructor crea un vuelo con un identificador 1, asigna la terminal1, el avión avion1, un destino en "Nueva York", una duración estimada de 5.5 horas, una duración real de 8.0 horas, una hora de salida a las 13:30 y la lista de pasajeros pasajeros1.

# Archivo .json

Los datos de los vuelos fueron almacenados exitosamente en un archivo .json por buenas prácticas.

## Ejemplo de la estructura del archivo.

Interfaz de usuario gráfica, Teams

Descripción generada automáticamente

Pantalla de computadora

Descripción generada automáticamente con confianza media

Pantalla de computadora

Descripción generada automáticamente