

Definición del Proyecto

El presente proyecto tiene como objetivo desarrollar un sistema de gestión de vuelos mediante la implementación de una estructura orientada a objetos, representada a través de un diagrama de clases UML. Este sistema busca modelar de forma precisa y organizada el funcionamiento interno de una aerolínea, abarcando la administración de vuelos, aviones, asientos y reservas, así como las relaciones entre los distintos componentes que intervienen en el proceso.

El desarrollo se fundamenta en los principios de la **Programación Orientada a Objetos (POO)**, aplicando conceptos clave como **composición, agregación, asociación y polimorfismo**, con el propósito de lograr una arquitectura modular, reutilizable y fácil de mantener.

Dentro del sistema, las clases principales cumplen roles específicos que contribuyen al correcto funcionamiento del modelo. Las clases **Económico, Ejecutivo y PrimeraClase** representan los distintos tipos de asientos disponibles en un avión, heredando características comunes de la clase abstracta **TipoAsiento**, lo que permite aplicar el principio de polimorfismo al definir comportamientos particulares según la categoría del asiento.

La clase **Asiento** se encarga de gestionar la información individual de cada asiento, incluyendo su número, ubicación, tipo y estado de ocupación.

Por su parte, la clase **EstadoAvion** describe las condiciones operativas del avión, tales como su ubicación, estado actual y capacidad de combustible, mientras que la clase **Avion** integra estos elementos y representa la estructura central del sistema de vuelos, relacionándose con las demás clases para reflejar la operación y administración de una aeronave dentro de la aerolínea.

El desarrollo del sistema se dividirá entre los integrantes del equipo. En este caso, la implementación de las clases **Económico, Ejecutivo, PrimeraClase, TipoAsiento, Asiento, EstadoAvion y Avion** será realizada por la estudiante María Alejandra Gonzales Cárdenas, mientras que su compañera se encargará de las clases restantes que completan el modelo general.

Finalmente, todas las clases se integrarán en un archivo principal denominado **main.py**, donde se llevará a cabo la **prueba y validación del funcionamiento general del sistema**, garantizando la coherencia entre los distintos módulos y el cumplimiento de los objetivos planteados.

Además, para extender la funcionalidad del sistema de gestión de vuelos y permitir la interacción con otros servicios o interfaces de usuario, se implementó una **API RESTful** utilizando el **framework Flask** en Python. Esta arquitectura asegura que la lógica de negocio, definida por el modelo de **Programación Orientada a Objetos (POO)**, esté disponible a través de *endpoints* bien definidos.

La **API** fue diseñada siguiendo un enfoque **Orientado a Recursos**, donde cada endpoint representa una entidad clave del negocio. Se buscó mantener la coherencia entre la estructura de las clases Python (**como Pasajero, Reserva, Vuelo, etc.**) y el formato de los datos intercambiados (**JSON**).

Se definieron e implementaron dos endpoints principales para gestionar las funcionalidades esenciales del sistema: Pasajeros y Reservas.

Pasajeros, este endpoint se encarga de la gestión de la información de los usuarios de la aerolínea, modelada por la clase Pasajero. Se implementaron los siguientes métodos **HTTP** para ofrecer una funcionalidad **CRUD (Crear, Leer, Actualizar, Eliminar)**.

Reservas, este endpoint gestiona el proceso central del sistema: la administración de las reservas. Las operaciones de este endpoint aprovechan la composición (relación **Reserva** con **Asiento**, **Vuelo**, y **Pasajero**) y el polimorfismo (al interactuar con las subclases de **TipoAsiento: Económico, Ejecutivo, PrimeraClase**).