

_5class_poo/class_vuelos.py

```
1  """
2  Imagina que estás desarrollando un sistema de reservas de vuelos para una aerolínea.
3  Crea un sistema de clases que permita a los usuarios realizar reservas de vuelos.
4  Aquí tienes una posible estructura:
5
6  - Clase base: `Vuelo`
7    - Atributos: número de vuelo, origen, destino, capacidad máxima, lista de pasajeros
8    - Métodos: agregar pasajero, verificar disponibilidad de asientos
9
10 - Clase derivada: `VueloEspecial` (hereda de `Vuelo`)
11   - Atributos adicionales: motivo del vuelo especial (por ejemplo, vacaciones, trabajo)
12
13 Resuelve el problema creando instancias de estas clases y realizando
14 reservas para diferentes vuelos y tipos de vuelos especiales.
15 """
16 # Definición de la clase base para vuelos
17 class Vuelo:
18     def __init__(self, numero, origen, destino, capacidad):
19         """
20         Constructor de la clase Vuelo.
21         """
22         self.numero = numero
23         self.origen = origen
24         self.destino = destino
25         self.capacidad = capacidad
26         self.pasajeros = [] # Lista para almacenar pasajeros
27
28     def verificar_disponibilidad(self):
29         """
30         Verifica la disponibilidad de asientos en el vuelo.
31         Retorna:
32             int: Número de asientos disponibles.
33         """
34         return self.capacidad - len(self.pasajeros)
35
36     def agregar_pasajero(self, pasajero):
37         """
38         Agrega un pasajero al vuelo, si hay asientos disponibles.
39         Parámetros:
40             pasajero (str): Nombre del pasajero a agregar.
41         """
42         if self.verificar_disponibilidad() > 0:
43             self.pasajeros.append(pasajero)
44             print(f"Pasajero {pasajero} agregado al vuelo {self.numero}")
45         else:
46             print("No hay asientos disponibles en este vuelo")
47
48 # Definición de la clase derivada para vuelos especiales
49 class VueloEspecial(Vuelo):
50     def __init__(self, numero, origen, destino, capacidad, motivo):
51         """
```

```
52     Constructor de la clase VueloEspecial.
53     """
54     # Llama al constructor de la clase base para inicializar atributos comunes
55     super().__init__(numero, origen, destino, capacidad)
56     self.motivo = motivo
57
58 # Ejemplo de uso de las clases
59 vuelo_regular = Vuelo("UA30", "NY", "MAD", 150)
60 vuelo_regular.agregar_pasajero("Juan")
61 print("Asientos disponibles:", vuelo_regular.verificar_disponibilidad())
62
63 vuelo_especial = VueloEspecial("UA31", "MIA", "BAR", 100, "Vacaciones")
64 vuelo_especial.agregar_pasajero("Laura")
65 print("Asientos disponibles en el vuelo especial:", vuelo_especial.verificar_disponibil←
66 idad())
67 print("Motivo del vuelo especial:", vuelo_especial.motivo)
67
```