



# UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

---

Documentazione Progetto:  
Object-Orientation  
**“Aeroporto”**

*Carmine Onorato N86005342*  
*Antonio Soritto N86004962*

**A.A. 2025**

---

# Indice

<b>1</b>	<b>Analisi traccia</b>	<b>2</b>
1.1	Traccia . . . . .	2
1.2	Identificazione delle entità . . . . .	2
1.3	Descrizione delle entità . . . . .	3
<b>2</b>	<b>Diagramma UML</b>	<b>5</b>

---

# 1 Analisi traccia

## 1.1 Traccia

Si sviluppi un sistema informativo per la gestione dell'aeroporto di Napoli, composto da una base di dati relazionale e da un applicativo Java con interfaccia grafica realizzata con Swing. Questo sistema deve consentire di organizzare e monitorare le operazioni aeroportuali in modo efficiente e intuitivo.

Il sistema può essere utilizzato da utenti autenticati tramite una login e una password. Gli utenti sono suddivisi in due ruoli: utenti generici, che possono prenotare voli, e amministratori del sistema, che gestiscono l'inserimento e l'aggiornamento dei voli.

Il sistema gestisce i voli in arrivo e quelli in partenza. Ogni volo è caratterizzato da un codice univoco, la compagnia aerea, l'aeroporto di origine (per i voli in arrivo a Napoli) e quello di destinazione (per i voli in partenza da Napoli), la data del volo, l'orario previsto, l'eventuale ritardo e lo stato del volo (programmato, decollato, in ritardo, atterrato, cancellato). Gli amministratori del sistema hanno la possibilità di inserire nuovi voli e aggiornare le informazioni sui voli esistenti.

Gli utenti generici possono effettuare prenotazioni per i voli programmati. Ogni prenotazione è legata a un volo e contiene informazioni come i dati del passeggero (che non deve necessariamente coincidere con il nome dell'utente che lo ha prenotato), il numero del biglietto, il posto assegnato e lo stato della prenotazione (confermata, in attesa, cancellata). Gli utenti possono cercare e modificare le proprie prenotazioni in base al nome del passeggero o al numero del volo.

Il sistema gestisce anche i gate di imbarco (identificati da un numero), assegnandoli ai voli in partenza. Gli amministratori possono modificare l'assegnazione dei gate.

Il sistema consente agli utenti di visualizzare aggiornamenti sui voli prenotati accedendo alla propria area personale, dove possono controllare eventuali ritardi, cancellazioni o variazioni direttamente dall'interfaccia. Inoltre, all'interno della homepage degli utenti viene mostrata una tabella con gli orari aggiornati dei voli in partenza e in arrivo, fornendo una panoramica immediata delle operazioni aeroportuali.

Infine, il sistema permette di eseguire ricerche rapide per trovare voli, passeggeri e bagagli in base a diversi criteri. Le informazioni più importanti vengono evidenziate, come i voli in ritardo o cancellati, per facilitare la gestione delle operazioni aeroportuali.

## 1.2 Identificazione delle entità

Seguendo la traccia fornita, possiamo identificare le seguenti entità:

- Utente

- 
- S.Utente(UtenteGenerico)
  - S.Utente(Amministratore)
  - Prenotazione
  - Enum(StatoPrenotazione)
  - Passeggero
  - Volo
  - Enum(StatoVolo)
  - S.Volo(VoloDestinazione)
  - S.Volo(VoloOrigine)
  - Gate

### 1.3 Descrizione delle entità

#### **Utente:**

Login, Password, VisualizzaVolo()

Descrizione:

Ogni utente possiede credenziali d'accesso e può visualizzare i voli programmati. Serve come superclass per Amministratore ed UtenteGenerico.

#### **S.Utente(UtenteGenerico)**

Nome, Prenotazione(), CercaPrenotazione\_Codice(), CercaPrenotazione\_Nome()

Descrizione:

Rappresenta un utente generico, tipicamente un passeggero, che può effettuare prenotazioni, cercarle tramite codice o tramite nome (Estende la classe Utente).

#### **S.Utente(Amministratore)**

Inserire(), Modifica(), AssegnaGate()

Descrizione:

Rappresenta un utente con privilegi che gli permettono di gestire ed amministrare ciò che concerne i voli (Estende la classe Utente).

#### **Prenotazione**

Posto, Stato, Numero, Bagaglio, CheckIn(), Prenotazione()

Descrizione:

Rappresenta le prenotazioni effettuata da un utente. Contiene informazioni sul posto, lo stato della prenotazione, il numero identificativo, i bagagli, e consente di eseguire il check-in.

---

### **Enum(StatoPrenotazione)**

-CONFERMATA, -IN\_ATTESA, -CANCELLATA

Descrizione:

Enumerazione che rappresenta lo stato di una prenotazione nel sistema.

### **Passeggero**

Nome, Cognome, IdDocumento

Descrizione:

Rappresenta un passeggero. I dati anagrafici e un identificativo del documento sono associati alle prenotazioni e ai voli.

### **Volo**

IdVolo, Compagnia, A\_Volo\_Destinazione, A\_Volo\_Origine, Data\_Volo, Ora\_Volo\_Prevista, Ritardo, stato

Descrizione:

Rappresenta un volo gestito dal sistema, con tutte le sue informazioni fondamentali: origine, destinazione, orari, compagnia, stato e ritardo

### **Enum(StatoVolo)**

-PROGRAMMATO, -IN\_RITARDO, -ATTERRATO, -DECOLLATO, -CANCELLATO

Descrizione:

Enumerazione che descrive lo stato attuale di un volo.

### **S.Volo(VoloDestinazione)**

setA\_Volo\_Destinazione()

Descrizione:

Classe focalizzata sulla gestione della destinazione del volo (Estende la classe Volo).

### **S.Volo(VoloOrigine)**

Imbarco

Descrizione:

Classe focalizzata sulla gestione dell'origine del volo (Estende la classe Volo).

### **Gate**

IdGate

Descrizione:

Classe focalizzata sulla gestione della destinazione del volo (Estende la classe Volo).

## 2 Diagramma UML

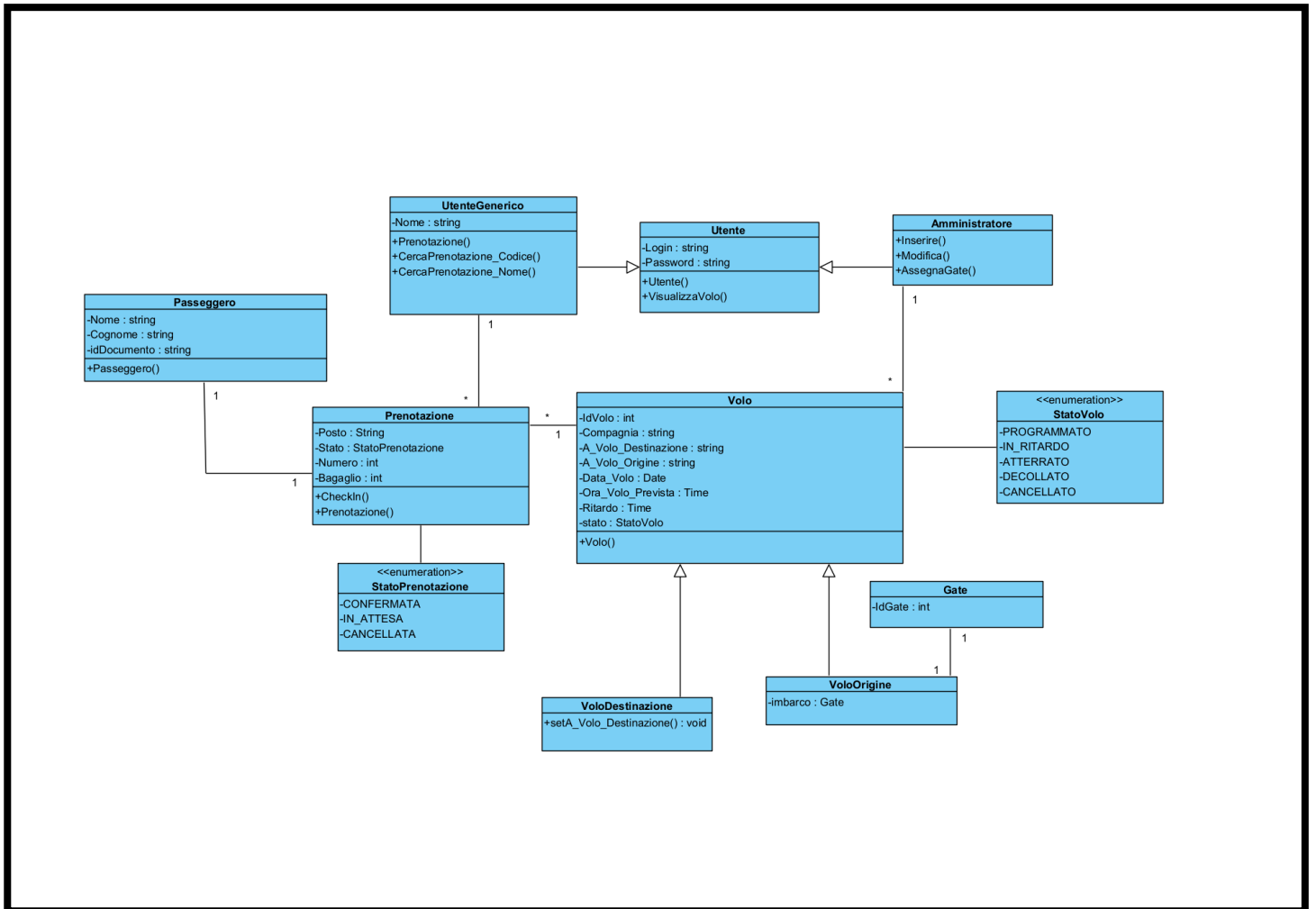


Figura 1: Diagramma UML