

# MVC

**Model-View-Controller (MVC)** è un pattern architettonale che divide un'applicazione in tre componenti principali per separare le responsabilità:

## **MODEL (Modello):**

- Contiene i dati e la logica di business
- Gestisce l'accesso al database
- Non sa nulla dell'interfaccia utente
- Esempio: la classe `Ticket` con i suoi attributi (km, level) e metodi (getPrice)

## **VIEW (Vista):**

- Si occupa della presentazione dei dati all'utente
- Mostra le informazioni provenienti dal Model
- Raccoglie gli input dell'utente
- Esempio: pagine HTML, interfacce grafiche, form per inserire i dati del biglietto

## **CONTROLLER (Controllore):**

- Fa da intermediario tra Model e View
- Riceve gli input dalla View
- Elabora le richieste e aggiorna il Model
- Decide quale View mostrare
- Esempio: gestisce la richiesta "calcola prezzo biglietto", prende i dati dalla View, chiama il Model e restituisce il risultato

## **Flusso tipico:**

1. L'utente interagisce con la View (esempio: inserisce km e classe)
2. La View invia la richiesta al Controller
3. Il Controller elabora i dati e aggiorna il Model

4. Il Model calcola/modifica i dati
5. Il Controller recupera i dati dal Model
6. Il Controller aggiorna la View con i nuovi dati
7. La View mostra il risultato all'utente

**Vantaggi:**

- Separazione delle responsabilità
- Codice più manutenibile e testabile
- Più sviluppatori possono lavorare contemporaneamente su componenti diverse
- Facile modificare l'interfaccia senza toccare la logica di business