



Corso di Laurea in Informatica - Università degli Studi di Napoli Federico II  
A.A. 2025/2026

# UninaFoodLab

**Calone Francesco N86005555**

**D'Angelo Mario N86005477**

Codice gruppo: **OOBD58**

Insegnamento di Basi di Dati I

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Descrizione del progetto . . . . .	2
1.2	Analisi dei Requisiti rilevanti per il Database . . . . .	2
<b>2</b>	<b>progettazione concettuale</b>	<b>3</b>
2.1	Introduzione . . . . .	3
2.2	UML non ristrutturato . . . . .	4
2.2.1	Entità principali . . . . .	5
2.2.2	Gerarchie e generalizzazioni . . . . .	5
2.2.3	Relazioni tra le entità . . . . .	5
2.2.4	Motivazione delle scelte progettuali . . . . .	6
2.3	UML ristrutturato . . . . .	7
2.3.1	Modifiche rispetto al modello non ristrutturato . . . . .	8
2.3.2	Comportamento del modello ristrutturato con le modifiche . . . . .	8
2.3.3	Motivazioni delle scelte del modello ristrutturato . . . . .	8

## 1 Introduzione

### 1.1 Descrizione del progetto

Il seguente progetto, denominato UninaFoodLab, nasce con l'obiettivo di progettare e implementare un sistema informativo per la gestione di corsi di cucina tematici. Il sistema si propone di supportare gli chef nella creazione e gestione di corsi articolati in sessioni teoriche e pratiche, facilitando al contempo l'iscrizione e la partecipazione degli utenti.

La documentazione che segue illustra in dettaglio tutte le fasi della progettazione, analizzando le scelte architetturali, i modelli concettuali e logici, e le soluzioni tecniche adottate per garantire la correttezza, la coerenza e l'efficienza del sistema.

### 1.2 Analisi dei Requisiti rilevanti per il Database

I seguenti requisiti funzionali sono stati analizzati ai fini della progettazione della base di dati. Essi definiscono le informazioni da memorizzare e le relazioni tra le entità del sistema.

1. Registrazione e autenticazione degli utenti.
  - Gli utenti devono poter creare un account con email e password.
  - Gli utenti devono poter effettuare il login.
  - Gli utenti devono poter recuperare la password dimenticata.
2. Gestione dei profili utente.
  - Gli utenti devono poter visualizzare e modificare le proprie informazioni personali.
  - Gli utenti devono poter visualizzare i propri corsi.
  - Gli utenti devono poter visualizzare i propri dati di pagamento.
  - Gli utenti devono poter visualizzare le proprie sessioni pratiche.
3. Creazione e gestione dei corsi da parte degli chef.
  - Gli chef devono poter creare nuovi corsi.
  - Gli chef devono poter definire le sessioni teoriche e pratiche.
  - Gli chef devono poter modificare o cancellare i corsi.
4. Iscrizione ai corsi da parte degli utenti.

- Gli utenti devono poter consultare l'elenco dei corsi disponibili.
- Gli utenti devono poter iscriversi a un corso.
- Gli utenti devono ricevere conferma dell'iscrizione.

5. Gestione delle presenze e dei pagamenti.

- Gli chef devono poter registrare la presenza alle sessioni.
- Il sistema deve gestire i pagamenti degli utenti.
- Gli utenti devono poter visualizzare lo stato dei loro pagamenti.

## 2 progettazione concettuale

### 2.1 Introduzione

Il modello concettuale rappresenta la struttura logica del database, definendo le entità, gli attributi e le relazioni tra di esse. In questa fase, si è proceduto a identificare le principali entità del sistema e a stabilire le relazioni che le collegano, garantendo così una visione chiara e coerente delle informazioni da gestire.

## 2.2 UML non ristrutturato

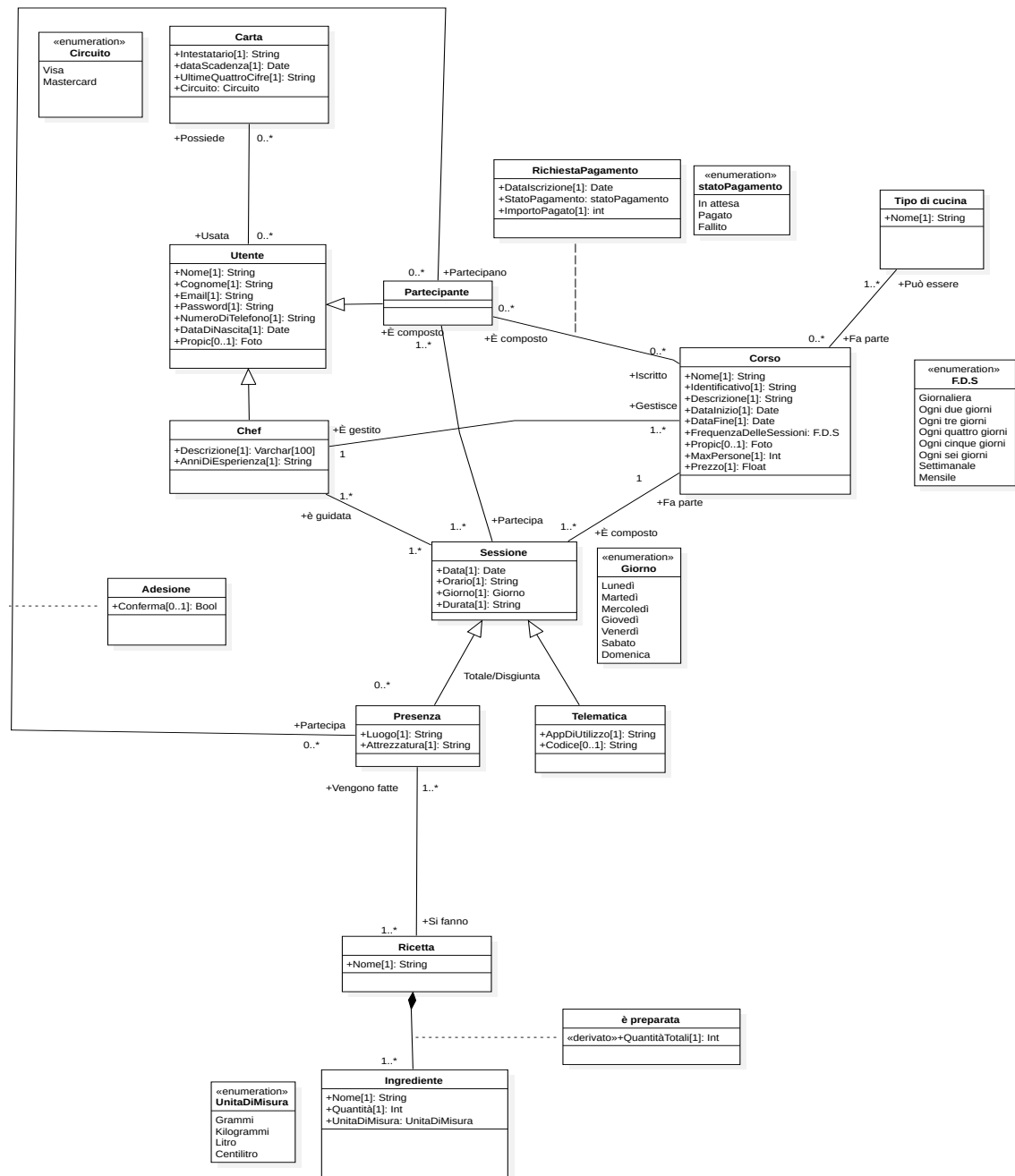


Figure 1: Diagramma UML del sistema

### 2.2.1 Entità principali

Le entità principali identificate nel sistema sono:

- **Utente:** L'utente rappresenta il soggetto fruitore del sistema, che può iscriversi ai corsi e partecipare alle sessioni. I principali attributi includono nome, cognome, email, password, telefono, data di nascita e una foto di profilo. Ogni utente può essere associato a una o più carte di pagamento e può diventare partecipante a diversi corsi.
- **Chef:** Lo chef è un utente con il ruolo specifico di organizzare corsi. Ogni chef dispone di una descrizione e di un numero di anni di esperienza. Un chef può gestire più corsi, ma ogni corso è gestito da un solo chef.
- **Corso:** Il corso è l'entità centrale del sistema e rappresenta una proposta didattica su un tema gastro-nomico specifico. Contiene attributi quali nome, descrizione, identificativo, data di inizio/fine, frequenza delle sessioni, prezzo, immagine di copertina e tipo di cucina (modellato come enumerazione). Ogni corso è composto da più sessioni e prevede una relazione molti-a-molti con i partecipanti.
- **Sessione:** Ogni corso è articolato in una o più sessioni, ciascuna delle quali ha una data, un orario, un insieme di giorni della settimana in cui si svolge, e una durata. Le sessioni sono specializzate in due sottotipi mutuamente esclusivi:
  - **Presenza:** Con attributi come luogo e attrezzature richieste.
  - **Telematica:** Con attributi relativi all'app utilizzata e al codice di accesso.
- **Partecipante e Adesione:** La partecipazione ai corsi è modellata tramite l'entità Partecipante, che collega utenti e corsi. La partecipazione a sessioni pratiche richiede un'adesione esplicita, rappresentata dall'entità Adesione, che contiene un attributo booleano di conferma.
- **Ricetta e Ingredientamento:** Ogni sessione pratica può includere la preparazione di una o più ricette. Ogni ricetta è composta da uno o più ingredienti, ciascuno dei quali ha un nome, una quantità e un'unità di misura (enumerata). La relazione tra Ricetta e Ingrediente è associativa e include l'attributo QuantitàTotale, utile per calcolare la quantità necessaria in base alle adesioni.
- **Carta e RichiestaPagamento:** Gli utenti possono associare al proprio profilo una o più carte di pagamento, appartenenti a un circuito specificato tramite enumerazione (Visa, Mastercard). Le richieste di pagamento sono entità separate, con data, stato (in attesa, pagato, fallito) e importo.

### 2.2.2 Gerarchie e generalizzazioni

Nel modello concettuale, sono state identificate le seguenti gerarchie e generalizzazioni:

- **Sessione:** Le sessioni sono suddivise in due sottotipi: *Presenza* e *Telematica*. Questa specializzazione consente di gestire le specificità di ciascun tipo di sessione, come il luogo e le attrezzature per le sessioni in presenza, e l'app utilizzata e il codice di accesso per quelle telematiche.
- **Utente:** L'entità Utente può essere specializzata in due sottotipi: *Partecipante* e *Chef*. Questa distinzione permette di gestire le diverse funzionalità e attributi associati a ciascun ruolo nel sistema.

Entrambe le specializzazioni sono totali e disgiunte, di conseguenza ogni istanza di Sessione sia esclusivamente di uno dei due tipi e che ogni Utente sia o un Partecipante o uno Chef, ma non entrambi contemporaneamente.

### 2.2.3 Relazioni tra le entità

Le relazioni tra le entità sono state definite come segue:

- **Utente - Partecipante:** Un utente può essere un partecipante a più corsi, e ogni corso può avere più partecipanti. Questa relazione è molti-a-molti.

- **Chef - Corso:** Ogni chef può gestire più corsi, ma ogni corso è associato a un solo chef. Questa relazione è uno-a-molti.
- **Corso - Sessione:** Un corso può avere più sessioni, ma ogni sessione appartiene a un solo corso. Questa relazione è uno-a-molti.
- **Sessione - Partecipante:** Ogni partecipante può aderire a più sessioni pratiche, e ogni sessione può avere più partecipanti. Questa relazione è multi-a-molti, mediata dall'entità Adesione.
- **Corso - Ricetta:** Ogni corso può includere più ricette, e ogni ricetta può essere associata a più corsi. Questa relazione è multi-a-molti.
- **Ricetta - Ingrediente:** Ogni ricetta può includere più ingredienti, e ogni ingrediente può essere utilizzato in più ricette. Questa relazione è multi-a-molti, mediata dall'attributo QuantitàTotale.
- **Utente - Carta:** Un utente può avere più carte di pagamento associate al proprio profilo. Questa relazione è uno-a-molti.
- **Ricetta - Ingrediente:** Ogni ricetta può essere associata a più ingredienti, e ogni ingrediente può essere utilizzato in più ricette. Questa relazione è una composizione, mediata dall'attributo QuantitàTotale.

#### 2.2.4 Motivazione delle scelte progettuali

Le scelte progettuali sono state guidate dalla necessità di garantire una rappresentazione chiara e coerente delle informazioni, facilitando la gestione dei corsi, delle sessioni e delle partecipazioni. La specializzazione delle sessioni in Presenza e Telematica consente di gestire le specificità di ciascun tipo di sessione, mentre la distinzione tra Partecipante e Chef permette di differenziare i ruoli degli utenti nel sistema. Inoltre, l'uso di relazioni multi-a-molti per gestire le adesioni alle sessioni pratiche e le associazioni tra ricette e ingredienti garantisce flessibilità e scalabilità nel modello.

## 2.3 UML ristrutturato

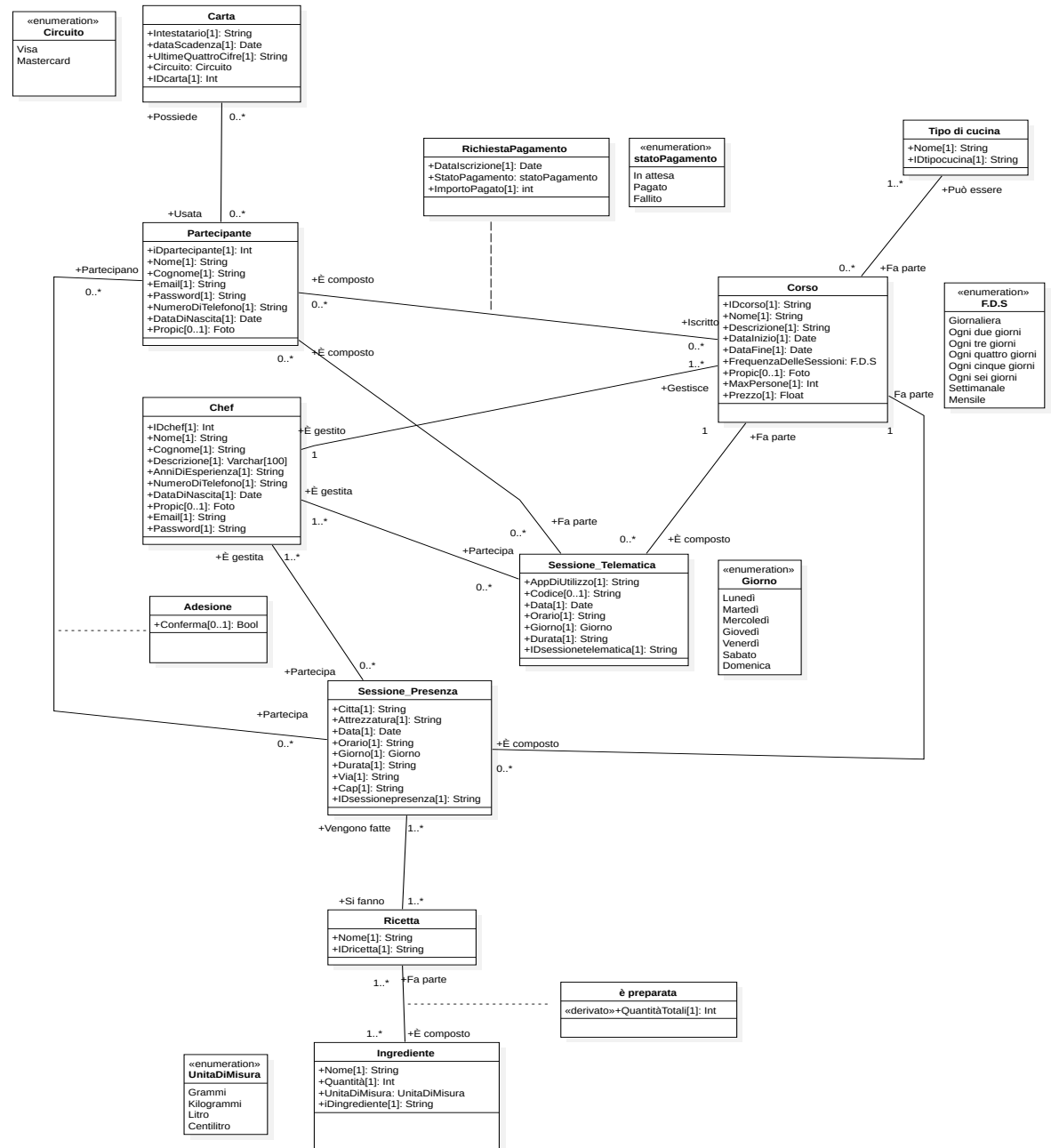


Figure 2: Diagramma UML ristrutturato del sistema

### 2.3.1 Modifiche rispetto al modello non ristrutturato

Il modello ristrutturato presenta alcune modifiche rispetto al modello non ristrutturato, adattandolo a un database relazionale. Le principali modifiche includono:

- **Rimozione delle gerarchie:** Le gerarchie tra le entità sono state rimosse, trasformando le specializzazioni in entità separate. Ad esempio, le sessioni di tipo Presenza e Telematica sono state trasformate in due entità distinte, mantenendo gli attributi specifici per ciascun tipo.
- **Attributi composti e multivalore:** Gli attributi composti e multivalore sono stati normalizzati. Ad esempio, l'attributo *Luogo* è stato suddiviso in attributi separati per *Città*, *Indirizzo* e *Cap*.
- **Aggiunta di chiavi primarie e esterne:** Ogni entità ha una chiave primaria univoca, e le relazioni tra le entità sono definite tramite chiavi esterne, garantendo l'integrità referenziale del database.

### 2.3.2 Comportamento del modello ristrutturato con le modifiche

Il modello ristrutturato mantiene le proprietà di specializzazione totale e disgiunta, adattandosi alle modifiche apportate. In particolare:

- **Specializzazione totale e disgiunta:** Le entità separate per le sessioni di tipo Presenza e Telematica garantiscono che ogni sessione appartenga esclusivamente a uno dei due tipi, rispettando la disgiunzione. Inoltre, ogni utente è classificato come Partecipante o Chef, assicurando che la specializzazione sia totale e disgiunta.
- **Integrità referenziale:** L'uso di chiavi primarie, esterne e l'aggiunta di chiavi surrogate garantisce che le relazioni tra le entità siano coerenti e che non si verifichino violazioni di integrità nel database.

### 2.3.3 Motivazioni delle scelte del modello ristrutturato

Le motivazioni alla base delle scelte progettuali del modello ristrutturato derivano dalla necessità di garantire una rappresentazione chiara, coerente e scalabile delle informazioni, adattandosi alle esigenze di un database relazionale. Una delle principali considerazioni riguarda la gestione delle specializzazioni totali e disgiunte, che hanno permesso di semplificare il modello concettuale senza perdere la coerenza logica.

Nel modello ristrutturato, le specializzazioni totali e disgiunte sono state gestite trasformando la classe generale in entità separate per ciascuna specializzazione. Questo approccio consente di incorporare gli attributi della classe generale direttamente nelle entità specializzate, eliminando la necessità di mantenere una gerarchia tra le entità. Ad esempio, nel caso delle sessioni, la classe generale "**Sessione**" è stata suddivisa in due entità distinte: "**Sessione\_Presenza**" e "**Sessione\_Telematica**". Ogni entità specializzata include gli attributi specifici del proprio tipo, come il luogo e le attrezzature richieste per le sessioni in presenza, e l'app utilizzata e il codice di accesso per quelle telematiche. In questo modo, si garantisce che ogni sessione appartenga esclusivamente a uno dei due tipi, rispettando la disgiunzione, e che tutte le sessioni siano rappresentate nel modello, rispettando la totalità, facilitando anche le associazioni tra le varie entità.

La stessa logica è stata applicata alla specializzazione dell'entità "Utente", suddivisa in "Partecipante" e "Chef". Ogni utente è classificato come Partecipante o Chef, ma non può appartenere a entrambe le categorie contemporaneamente. Gli attributi comuni agli utenti, come nome, cognome, email e data di nascita, sono stati incorporati direttamente nelle entità specializzate, mentre gli attributi specifici, come la descrizione e gli anni di esperienza per gli Chef, sono stati aggiunti solo alla rispettiva entità. Questo approccio semplifica la gestione delle entità nel database, eliminando la necessità di un'entità generale "Utente" e garantendo che la specializzazione rimanga totale e disgiunta.

Per quanto riguarda le associazioni tra le entità, il modello ristrutturato utilizza chiavi primarie e chiavi esterne per definire le relazioni, garantendo l'integrità referenziale del database. Ad esempio, la relazione tra "Corso" e "Sessione" è uno-a-molti, con l'identificativo del corso utilizzato come chiave esterna nelle entità "Presenza" e "Telematica". Questo approccio consente di mantenere la coerenza delle informazioni e di semplificare le operazioni di query e aggiornamento. Analogamente, la relazione molti-a-molti tra "Sessione" e "Partecipante" è mediata dall'entità "Adesione", che include un attributo booleano di conferma per gestire la partecipazione alle sessioni pratiche.



Un altro esempio significativo riguarda la relazione tra "Ricetta" e "Ingrediente". Nel modello ristrutturato, questa relazione è stata normalizzata utilizzando un'entità associativa che include l'attributo "QuantitàTotale". Questo approccio consente di calcolare facilmente la quantità necessaria di ciascun ingrediente in base alle adesioni alle sessioni pratiche, garantendo flessibilità e scalabilità nel modello.

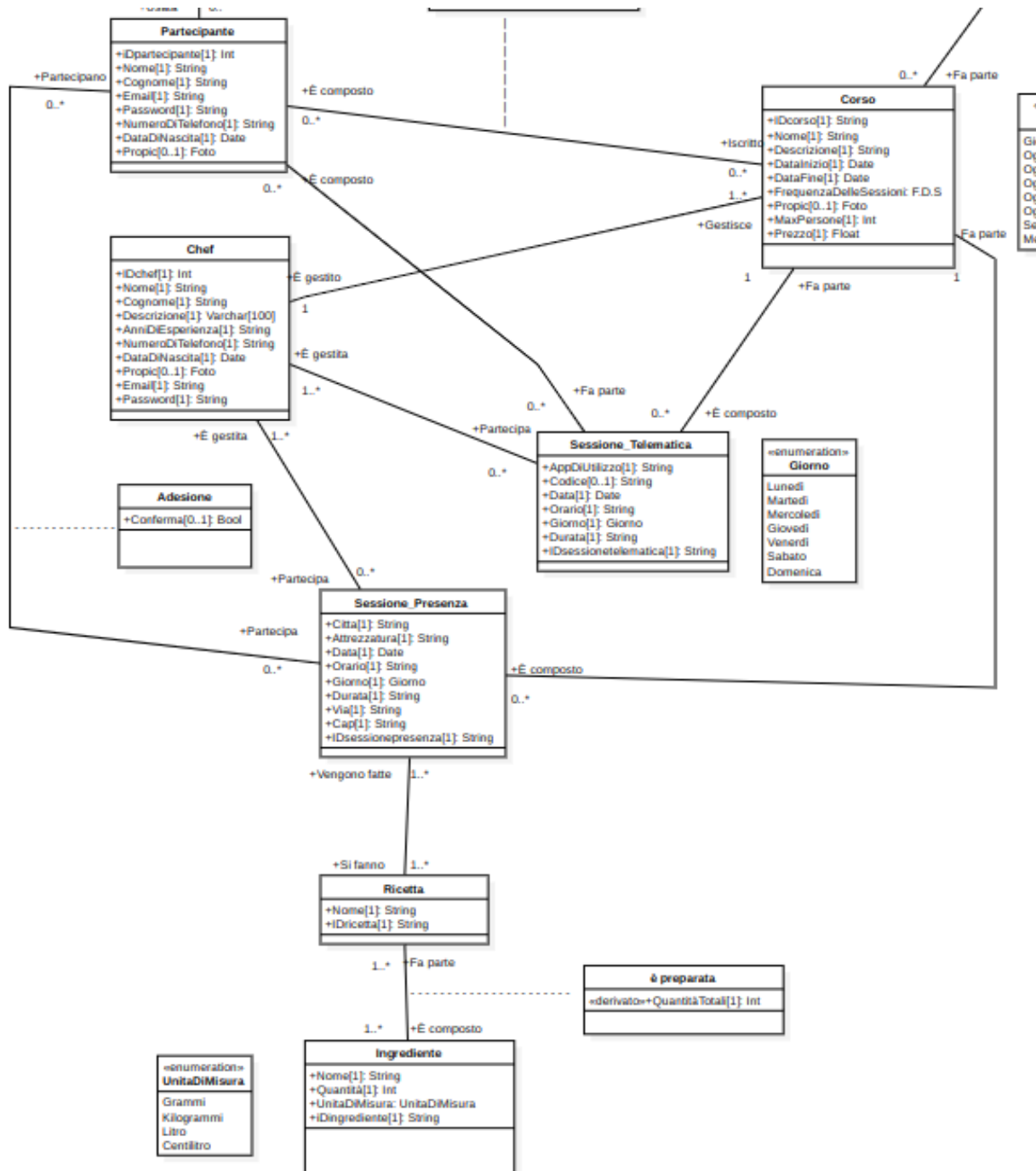


Figure 3: Scelte ristrutturato