

1. SCELTA DI GENERALIZZARE UTENTE

Abbiamo deciso di introdurre una generalizzazione per la classe `Utente` al fine di gestire in modo più chiaro e flessibile le due principali funzionalità del sistema: seguire un corso (nel ruolo di studente) e gestire un corso (nel ruolo di chef).

Questa scelta consente di modellare correttamente il comportamento degli utenti che possono ricoprire ruoli diversi all'interno della piattaforma, senza duplicare informazioni o relazioni.

È importante sottolineare che la generalizzazione è di tipo totale, poiché non possono esistere utenti generici che non siano almeno uno dei due ruoli previsti, e overlap, poiché i ruoli possono sovrapporsi. In particolare, uno chef può anche iscriversi e partecipare ai corsi organizzati da altri chef, mantenendo così la coerenza con le dinamiche reali del dominio applicativo.

2. SCELTA DI GENERALIZZARE SESSIONE

Abbiamo deciso di adottare una generalizzazione per la classe `Sessione`, con lo scopo di separare e gestire in modo più efficiente le funzionalità specifiche delle sessioni in presenza rispetto a quelle online. Questa scelta consente di rappresentare in maniera più chiara le differenze operative tra i due tipi di sessione, in particolare per quanto riguarda la gestione delle adesioni e del numero di partecipanti, che riguardano esclusivamente le sessioni pratiche in presenza.

La generalizzazione è definita come totale, poiché ogni sessione deve appartenere necessariamente a una delle due sottoclassi (`SessionePresenza` o `SessioneOnline`), e disgiunta, in quanto il sistema non prevede la possibilità di una modalità ibrida o mista. Tale modellazione garantisce una corretta rappresentazione del dominio, evitando ambiguità e semplificando la gestione delle diverse tipologie di sessioni.

3. SCELTA DI USARE L'ATTRIBUTO DERIVATO /NumeroAderenti

Abbiamo deciso di introdurre l'attributo derivato `/NumeroAderenti` all'interno della classe `SessionePresenza`, al fine di evitare la memorizzazione ridondante di tale informazione. Il suo valore non viene quindi archiviato direttamente, ma calcolato dinamicamente attraverso un'operazione di *count* sulle istanze della classe associativa `Adesione`, che collega le sessioni pratiche agli utenti partecipanti.

Questa scelta consente di mantenere il modello più coerente ed efficiente, riducendo la possibilità di inconsistenze nei dati e migliorando la normalizzazione. Inoltre, l'attributo `/NumeroAderenti` riveste un ruolo centrale ai fini del progetto, in quanto il suo valore viene utilizzato per determinare un secondo attributo derivato, `/QuantitàTotale`, necessario per stimare la quantità complessiva di ingredienti da impiegare in ciascuna sessione e per ottimizzare la gestione delle risorse alimentari, riducendo così gli sprechi.

4. SCELTA DI USARE L'ATTRIBUTO DERIVATO /QuantitàTotale

Abbiamo deciso di introdurre l'attributo derivato /QuantitàTotale all'interno della classe associativa SessioneIngrediente, con l'obiettivo di evitare la memorizzazione ridondante di questo dato e garantire un calcolo dinamico in fase di esecuzione. Il valore dell'attributo non viene quindi salvato esplicitamente, ma calcolato automaticamente moltiplicando l'attributo derivato /NumeroAderenti, presente nella classe SessionePresenza, per l'attributo QuantitàNecessaria, definito nella classe associativa RicettaIngrediente.

Questa soluzione consente di mantenere la coerenza e la normalizzazione dei dati, riducendo il rischio di anomalie e semplificando la gestione delle informazioni. Inoltre, l'attributo /QuantitàTotale rappresenta un elemento chiave del sistema, poiché costituisce il fulcro della funzionalità dedicata alla riduzione dello spreco alimentare, permettendo di stimare in modo accurato la quantità complessiva di ingredienti necessari per ogni sessione pratica in base al numero effettivo di partecipanti.

5. SCELTA DI CREARE ENUMERAZIONE SpecializzazioneChef E TipologiaArgomento

Agli attributi Specializzazione e Argomento si è scelto di assegnare, rispettivamente, i tipi enumerativi <<enum>> SpecializzazioneChef e <<enum>> TipologiaArgomento, anziché un generico tipo String.

Questa scelta è stata adottata per evitare ambiguità e limitare l'inserimento di valori non coerenti con il dominio applicativo, prevenendo così la creazione di corsi non attinenti al contesto culinario.

Inoltre, tale modellazione introduce un vincolo semantico tra i due attributi: uno *Chef* potrà gestire e creare soltanto corsi coerenti con la propria specializzazione.

Ad esempio, uno chef con specializzazione *fornaio* potrà proporre corsi di *panetteria*, ma non corsi di *pasticceria* o *cucina asiatica*.

In questo modo si garantisce una maggiore coerenza tra le competenze degli chef e gli argomenti trattati nei corsi.

RISTRUTTURAZIONE

1. ELIMINAZIONE DELLE GERARCHIE

Nel modello iniziale era presente una generalizzazione della classe Utente nelle sottoclassi Studente e Chef. Abbiamo deciso di adottare la seconda strategia, accorpendo le due sottoclassi all'interno della classe padre Utente. Questa scelta è motivata dall'assenza di attributi propri nella classe Studente, che avrebbe reso la distinzione strutturale poco significativa. Per mantenere comunque la distinzione dei ruoli, è stato introdotto un attributo discriminante TipoUtente, collegato all'enumerazione <<enum>>TipologiaUtente, i cui valori possibili sono *Chef*, *Studente* e *ChefStudente*. Quest'ultimo valore è stato aggiunto per rappresentare i casi

in cui un utente possa ricoprire entrambi i ruoli contemporaneamente, in coerenza con la natura overlap della generalizzazione originaria.

Analogamente, nel modello iniziale era presente una generalizzazione della classe *Sessione* nelle sottoclassi *SessionePresenza* e *SessioneOnline*. Anche in questo caso è stata adottata la strategia di accorpamento nella classe padre, in quanto la classe *SessioneOnline* offriva un contributo informativo limitato e non tale da giustificare una modellazione separata. È stato pertanto introdotto un attributo discriminante *TipoSessione*, che fa riferimento all'enumerazione <<enum>>*TipologiaSessione*, composta da due valori: *Online* e *Presenza*. Questo attributo consente di gestire correttamente la natura disgiunta della generalizzazione e di distinguere in modo chiaro le diverse modalità di erogazione delle sessioni.

2. SCELTA DEGLI IDENTIFICATORI PRIMARI

All'interno delle varie classi è stato introdotto un identificatore distintivo (chiave surrogata), contraddistinto dal prefisso *Id*, con lo scopo di garantire l'univocità delle istanze e semplificare la gestione delle relazioni nella fase logica del modello.

Fa eccezione la classe *Utente*, nella quale si è scelto di utilizzare l'attributo *Email* come chiave naturale, in quanto già univoca per ogni utente del sistema.

Per quanto riguarda invece le classi associative *Adesione*, *RicettaIngrediente* e *SessioneIngrediente*, non è stata introdotta una chiave surrogata, ma si è preferito adottare una chiave composta, formata dagli identificatori distintivi delle entità coinvolte nella relazione.

Questa scelta consente di mantenere la coerenza logica del modello, evitando ridondanze e rispettando la natura delle associazioni molte-a-molte da cui tali classi derivano.