

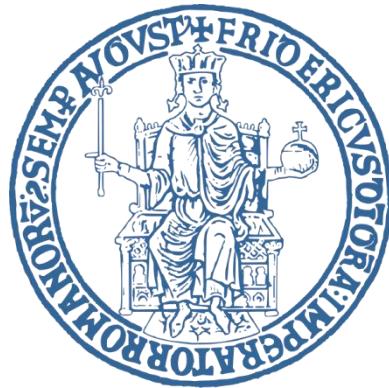


DOCUMENTAZIONE

Object Orientation



UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II



COSO DI LAUREA IN INFORMATICA

ANNO ACCADEMICO 2025/2026

Object Orientation

FoodLab Unina 2025

A cura di:

Correra Salvatore

N86005207

s.correra@studenti.unina.it

Amendola Pierfrancesco

N86005039

pi.amendola@studenti.unina.it

De Benedictis Umberto

N86005148

u.debenedictis@studenti.unina.it

Codice gruppo: OOBDB8

UNINAFOODLAB

Indice

Capitolo 1: Introduzione

Sommario

FoodLab Unina 2025.....	1
Indice.....	2
 1.1 Richiesta del committente.....	3
 1.2 Breve descrizione dell'Applicazione	3
 1.3 Panoramica delle Funzionalità.....	4
 1.4 Entità	5
 1.5 Tecnologie Utilizzate	5
 2.1 Pattern Architetturale: MVC.....	6
MainApplication.java (Il Container):	6
Controller.java (Il Cervello):	6
KitchenTheme.java (Il Design System):.....	7
DAO e Entity:.....	7
 2.2 Descrizione delle entità del problema.....	8
Class Diagram entità	9
 2.4 Control: descrizione del Controller.....	10
 2.5 Descrizione delle Interfacce.....	13
 2.6 Class Diagram del Sistema	19

Capitolo 1:

Introduzione

1.1 Richiesta del committente

Si sviluppi un applicativo Java con interfaccia grafica (Swing o JavaFX) per la gestione dei corsi tematici offerti dalla piattaforma UninaFoodLab. Il sistema dovrà essere collegato a un database relazionale prepopolato contenente informazioni su chef, ricette e ingredienti.

Il sistema deve permettere l'autenticazione degli chef tramite credenziali (username e password). Una volta autenticato, lo chef può aggiungere un nuovo corso, specificando le seguenti informazioni: categoria, data di inizio, frequenza delle sessioni, numero di sessioni. Per ciascuna sessione, deve essere indicata la modalità di svolgimento, ovvero se si tratta di una sessione online o in presenza.

Lo chef avrà inoltre la possibilità di visualizzare i corsi esistenti, applicando filtri per categoria. Dopo aver selezionato un corso, lo chef può associare a ciascuna sessione pratica una o più ricette da realizzare.

Infine, il sistema deve fornire un report mensile, che permette allo chef di visualizzare: il numero di corsi totali tenuti, il numero di sessioni online e pratiche, e di quest'ultime il numero medio, massimo e minimo di ricette realizzate. Il report deve fornire una rappresentazione grafica dei dati, realizzata utilizzando una libreria come JFreeChart.

1.2 Breve descrizione dell'Applicazione

UninaFoodLab – Chef Dashboard è un'applicazione desktop con interfaccia in Java Swing, progettata per gli chef di un laboratorio culinario. Fornisce una dashboard centralizzata per gestire le proprie attività didattiche, tra cui la creazione di nuovi corsi di cucina, la gestione delle singole sessioni e l'associazione di ricette specifiche alle sessioni pratiche. L'applicazione offre anche strumenti di reportistica per monitorare le attività svolte in un dato periodo di tempo.

1.3 Panoramica delle Funzionalità

- **Autenticazione:** Accesso al sistema tramite credenziali (username o email/password). Inoltre è permessa la registrazione di nuovi chef o studenti
- **Gestione Corsi:** Creazione di nuovi corsi, specificando categoria, data di inizio, numero e frequenza delle sessioni. È possibile anche eliminare dei corsi
- **Gestione Sessioni:** Definizione della modalità (online o in presenza) per ogni sessione di un dato corso.
- **Gestione Ricette:** Creazione ed eliminazione delle ricette, ciascuna ricetta è associata ad una lezione.
- **Visualizzazione e Filtro:** Elenco di tutti i corsi tenuti dallo chef, con possibilità di filtrare per categoria.
- **Reportistica:** Generazione di un report mensile con statistiche aggregate e una visualizzazione grafica.
- **Sezione Chef:** Sezione destinata solo agli chef, dove una volta autenticati, si accede alla panoramica e alle azioni che uno chef può realizzare
- **Sezione Studente:** Sezione destinata solo agli studenti, dove una volta autenticati, si accede alla panoramica e alle azioni che uno studente può effettuare. Lo studente può iscriversi alle lezioni create dagli chef e consultare le notifiche
- **Gestione notifiche:** Lo chef quando crea una sessione, modifica una sessione, aggiunge una ricetta, il sistema invia una notifica diretta allo studente. Quando uno studente che si iscrive ad un corso, il sistema invia una notifica diretta allo chef che tiene quel corso
- **Visualizzazione informazioni sui corsi:** Lo chef può visualizzare i dettagli di ciascun corso, ovvero i dettagli di tutte le sessioni, delle loro ricette e degli ingredienti richiesti da ciascuna

1.4 Entità

- **Utente**
- **Corso**
- **Iscrizione**
- **Sessione**
- **Ricetta**
- **Avviso**
- **NotificaChef**
- **NotificaCorso**
- **ReportMensile**

1.5 Tecnologie Utilizzate

- **Linguaggio:** Java 21
- **GUI:** Java Swing
- **Grafici:** JFreeChart
- **DBMS:** PostgreSQL
- **Class Diagram:**
- **Version Control:** <https://github.com/Iamtheflash-four/FoodLab2025.git>

2.1 Pattern Architetturale: MVC

Il progetto segue il pattern architetturale **MVC (Model-View-Controller)**, che garantisce una netta separazione tra i dati, l’interfaccia utente e la logica di business.

- **Model (Modello):** Rappresenta i dati e la logica di accesso ai dati.
 - **Package entity:** Contiene le classi che mappano le tabelle del database (es. Utente, Corso, Sessione). Sono semplici POJO (Plain Old Java Objects).
 - **Package DAO (Data Access Object):** Gestisce la comunicazione diretta con il database PostgreSQL (es. ChefDAO, CorsoDAO). Ogni classe DAO si occupa delle query SQL per una specifica entità.
- **View (Vista):** Rappresenta l’interfaccia grafica.
 - **Package boundary:** Contiene i pannelli Swing (es. AreaChefPanel, LoginPanel) e la classe principale MainApplication.
- **Controller (Controllore):** Gestisce l’interazione tra Model e View.
 - **Package controller:** La classe Controller funge da intermediario. Riceve gli input dalla View (es. click su “Login”), chiama il Model (DAO) per i dati e aggiorna la View.

Classi Principali e Scelte Progettuali:

MainApplication.java (Il Container):

- **Scelta:** Utilizzo di un JFrame singolo con **CardLayout**.
- **Perché?**: Invece di aprire e chiudere finestre multiple (che può essere confusionario), CardLayout permette di “sfogliare” le schermate come carte nello stesso frame. Questo rende la navigazione fluida e moderna.
- **Funzionamento:** Inizializza tutti i pannelli (Paginaliniziale, Login, ecc.) e permette di passare da uno all’altro tramite metodi come showCard("NOME_CARD").

Controller.java (Il Cervello):

- **Scelta:** Un controller centralizzato passato a tutte le viste.
- **Perché?**: Centralizza la logica di business. Le viste non parlano mai direttamente col database, ma chiedono al Controller di farlo. Questo migliora la manutenibilità e la sicurezza.

- **Feature Notabile:** Il metodo analyzeConnectionError intercetta le eccezioni SQL e restituisce messaggi d'errore “umani” (es. suggerisce di avviare Postgres se la connessione è rifiutata), molto utile per il debug.

KitchenTheme.java (Il Design System):

- **Scelta:** Una classe statica che definisce colori, font e componenti custom.
- **Perché?:** Garantisce coerenza grafica (Look & Feel). Invece di impostare i colori in ogni pannello, si usano costanti come KitchenTheme.WARM_ORANGE.
- **Componenti Custom:** Metodi come createKitchenButton o createKitchenTextField creano componenti Swing già stilizzati con bordi arrotondati e colori del tema, evitando duplicazione di codice grafico.

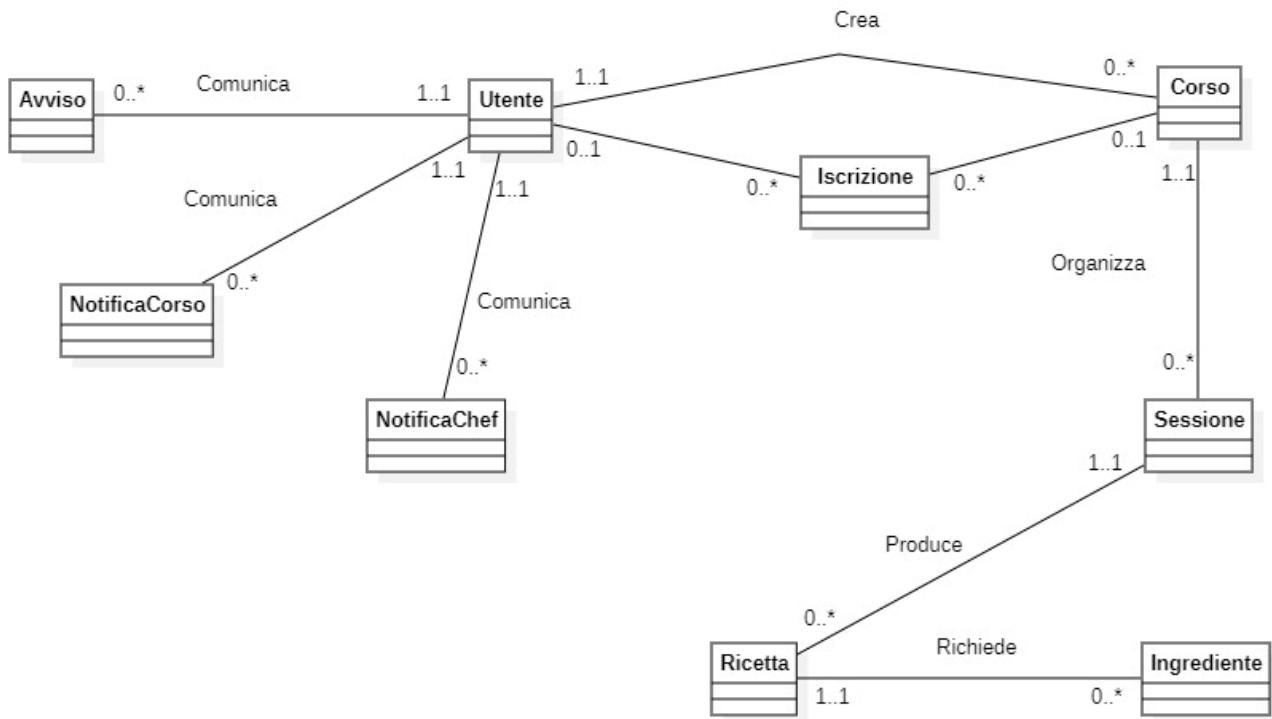
DAO e Entity:

- **Scelta:** Pattern DAO.
- **Perché?:** Isola le query SQL dal resto del codice. Se domani cambiassimo database, dovremmo modificare solo i DAO e non l'interfaccia grafica.

2.2 Descrizione delle entità del problema

- **Utente:** rappresenta la persona che utilizzerà l'applicazione, l'utente può essere uno studente che utilizza l'applicazione per iscriversi e frequentare i corsi di cucina. Può anche avere dei privilegi di uno chef che può creare dei corsi, modificarli ed eliminarli
Dell'utente si memorizza il codice fiscale, nome, cognome, la data di nascita, l'username, l'email e la password.
- **CORSO:** rappresenta un corso di cucina. Un corso è gestito da uno chef che può, una volta creato, inserire delle sessioni.
Si memorizzano un codice identificativo, l'argomento, la descrizione, data inizio, data fine, il numero massimo dei partecipanti, frequenza e la modalità di frequenza preferita del corso (online o in presenza)
- **Iscrizione:** rappresenta l'iscrizione di un utente ad un corso tenendo traccia dell'identificativo del corso e del codice fiscale dell'utente
- **Sessione:** lezioni di cucina tenute dallo chef, possono essere svolte online o in presenza. Le sessioni in presenza possono produrre delle ricette.
Si memorizzano un numero identificativo della sessione, la data d'inizio, l'ora di inizio, la modalità di svolgimento
- **Ricetta:** rappresenta una ricetta, è prodotta da una sessione in presenza, richiede ingredienti che gli chef dovranno inserire indicando la quantità
- **Avviso:** rappresenta un avviso che viene scritto da uno chef o generato automaticamente in caso creazione, spostamento o annullamento di una sessione
- **NotificaChef:** rappresenta una notifica destinata agli Chef del sistema FoodLab. Viene utilizzata per informare lo Chef di eventi rilevanti relativi ai propri corsi come ad esempio iscrizioni, promemoria o cancellazioni di iscrizioni.
- **NotificaCorso:** rappresenta una richiesta di notifica da parte di uno Studente per essere avvisato quando un corso diventa disponibile o quando ci sono aggiornamenti.
- **ReportMensile:** rappresenta report mensile che fornisce il numero di corsi totali tenuti, il numero di sessioni online e pratiche, e di quest'ultime il numero medio, massimo e minimo di ricette realizzate da uno chef

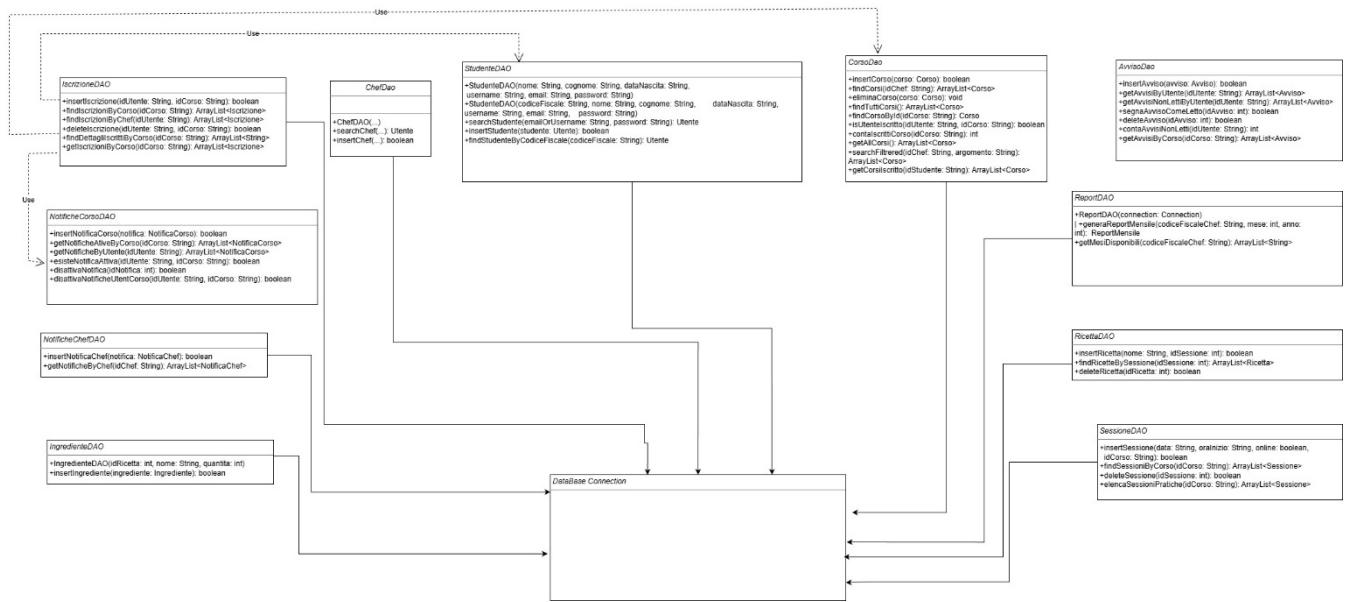
Class Diagram entità



2.3 Descrizione dei DAO

- **ChefDAO**: Responsabile della gestione dei dati degli chef/insegnanti che tengono i corsi. Oltre a recuperare le informazioni anagrafiche degli chef, gestisce anche l'autenticazione tramite username e password. Carica in modo completo il profilo dello chef includendo tutti i corsi da lui tenuti.
- **CorsoDAO**: Si occupa di creare nuovi corsi e recuperarli con tutti i dati associati (categoria, frequenza, chef, sessioni), permettendo filtraggio per chef e categoria. Coordina il caricamento di tutte le informazioni correlate per fornire una vista completa del corso.
- **SessioneDAO**: Si occupa sia della creazione di nuove sessioni che del loro recupero completo di tutte le ricette associate.
- **RicettaDAO**: Gestisce il catalogo delle ricette culinarie presenti nel sistema. Permette di recuperare le ricette sia individualmente che in relazione alle sessioni in cui vengono trattate, fornendo i dati necessari per associare contenuti didattici specifici alle lezioni.
- **SudenteDAO**: permette di registrare, autenticare e gli studenti oppure recuperare i dati di uno studente registrato nel sistema, partendo dal suo codice fiscale

- **IscrizioneDAO:** permette di registrare ed eliminare le iscrizioni di ciascun studente nel database. Può inoltre recuperare tutte le iscrizione di un corso o di uno studente
- **ReportDAO:** recupera dal database tutte le informazioni per costruire un ReportMensile per uno chef



Ogni classe DAO è responsabile della gestione della sua corrispondente classe entità ed è ovviamente in relazione con quest'ultima.

Per motivi di chiarezza visiva, sono state omesse le classi entità al fine di mostrare solo in che modo i DAO interagiscono tra loro.

2.4 Control: descrizione del Controller

Il Controller agisce come il cervello dell'applicazione: riceve input dalla vista, elabora le richieste utilizzando il modello e determina quale vista mostrare successivamente.

In particolare, il Controller si occupa di:

1. **Gestione della Logica di Business:** Contiene tutta la logica operativa dell'applicazione. Ad esempio, gestisce l'autenticazione dello chef, la creazione di nuovi corsi, l'associazione di ricette a una sessione, la generazione di report.

2. **Orchestrazione dei Dati:** Utilizza i DAO per recuperare, salvare e modificare i dati nel database.
3. **Controllo del flusso di esecuzione:** Gestisce la navigazione tra le diverse schermate. Ad esempio, dopo un login corretto, chiude la LoginPage e mostra la MainPage.
4. **Gestione dello Stato:** Mantiene in memoria lo stato corrente dell'applicazione, come lo chef che ha effettuato l'accesso (chefAutenticato) e il corso attualmente selezionato dall'utente (corsoSelezionato).
5. **Ponte tra View e Model:** Risponde alle interazioni dell'utente catturate dalla View (es. un click su un bottone), esegue le operazioni necessarie interagendo con il Model e, infine, aggiorna la View con i nuovi dati.

Controller
<pre> registraChef(Utente chef): void effettuaLogin(String emailOrUsername, String password): void searchStudente(String emailOrUsername, String password): Utente insertStudente(Utente nuovoStudente): boolean logout(): void creaCorso(Corso corso): void elencaCorsi(String idChef): ArrayList<Corso> eliminaCorso(Corso corso): void getAllCorsi(): ArrayList<Corso> searchFiltrered(String codiceFiscale, String argomento): ArrayList<Corso> getCorsiscritto(String codiceFiscale): ArrayList<Corso> contalscrittiCorso(String idCorso): int isUtentelscritto(String codiceFiscale, String idCorso): boolean creaSessione(String data, String orainizio, boolean online, String idCorso): void elencaSessioni(String idCorso): ArrayList<Sessione> eliminaSessione(int idSessione): void elencaSessioniPratiche(String idCorso): ArrayList<Sessione> getNumeroSessioniCorso(String idCorso): int creaRicetta(String nome, int idSessione): void elencaRicette(int idSessione): ArrayList<Ricetta> eliminaRicetta(int idRicetta): void getRicetteByCorso(String idCorso): ArrayList<Ricetta> insertIngrediente(Ingrediente ingrediente): void getIngredientiByRicetta(int idRicetta): ArrayList<Ingrediente> deleteIngrediente(int idRicetta, String nomeIngrediente): void iscriviChef(String idChef, String idCorso): void insertIscrizione(String codiceFiscale, String idCorso): boolean elencaIscrizioniCorso(String idCorso): ArrayList<Iscrizione> elencaIscrizioniChef(String idChef): ArrayList<Iscrizione> getIscrizioniByCorso(String courself): ArrayList<Iscrizione> disiscriviChef(String idChef, String idCorso): void getNumerolscrittiCorso(String idCorso): int getDettagliIscrittiCorso(String idCorso): ArrayList<String> insertAvviso(Avviso avviso): boolean getAvvisiByUtente(String codiceFiscale): ArrayList<Avviso> getAvvisiByCorso(String idCorso): ArrayList<Avviso> segnaAvvisoComeLetto(int idAvviso): void getNotificheByChef(String codiceFiscale): ArrayList<NotificaChef> segnaNotificaComeLetta(int idNotifica): boolean esisteNotificaAttiva(String codiceFiscale, String idCorso): boolean insertNotificaCorso(NotificaCorso notifica): boolean getMesiDisponibili(String codiceFiscaleChef): ArrayList<String> generaReportMensile(String codiceFiscaleChef, int mese, int anno): ReportMensile showChefLogin(): void showAreaChef(): void showGestioneCorsi(): void showCreazioneCorsi(): void showNotificheChef(): void showReportMensile(String codiceFiscaleChef): void getConnection(): Connection </pre>

2.5 Descrizione delle Interfacce

L'applicazione FoodLab adotta un **tema ispirato al mondo della cucina**, con una paletta di colori elegante e accogliente che richiama l'atmosfera di una cucina professionale. Ogni interfaccia è progettata per essere **intuitiva e funzionale**, comunicando esclusivamente con il Controller per eseguire operazioni e recuperare dati.

Pagina Iniziale

La schermata di benvenuto presenta il logo dell'applicazione e offre due percorsi principali:

- Accesso Chef: per i professionisti della cucina che gestiscono i corsi
- Accesso Studente: per gli appassionati che desiderano iscriversi ai corsi

Area Riservata Chef

Il pannello principale dello Chef funge da hub centrale per tutte le operazioni:

- Dashboard personalizzata con riepilogo dei corsi attivi
- Menu di navigazione verso le diverse funzionalità:
 - Gestione Corsi
 - Notifiche
 - Report Mensili
 - Creazione nuovi corsi
- Visualizzazione rapida delle statistiche e degli avvisi recenti

Gestione Corsi

Interfaccia completa per la gestione dei corsi culinari con:

- Lista corsi: visualizzazione di tutti i corsi creati dallo Chef
- Dettagli corso: informazioni complete su ogni corso selezionato
- Funzionalità di modifica:
 - Aggiunta e gestione di ricette (tramite RecipeDialog)

- Inserimento di ingredienti con quantità e unità di misura (tramite IngredientDialog)
- Creazione di sessioni con date e orari (tramite CreateSessionDialog)
- Pubblicazione di avvisi per gli studenti iscritti (tramite CreateAnnouncementDialog)
- Filtri di ricerca per trovare rapidamente i corsi desiderati (FiltroPanel)

Notifiche Chef

Centro notifiche dedicato che mostra:

- Nuove iscrizioni ai corsi
- Messaggi dagli studenti
- Promemoria per sessioni imminenti
- Sistema di notifiche lette/non lette con indicatori visivi

Report Mensile

Pannello analitico per monitorare le performance:

- Statistiche su iscrizioni e partecipazioni
- Grafici riassuntivi dell'andamento mensile
- Esportazione dati per analisi approfondite
- Visualizzazione dei trend e confronto tra periodi



FoodLab Unina

Dove la passione per la cucina prende vita

Benvenuti nella nostra cucina digitale!

Scegli il tuo ruolo e inizia la tua avventura culinaria



CHEF

Sei un esperto culinario?

Accedi alla tua area riservata per creare e gestire i tuoi corsi di cucina

[Accedi come Chef](#)



STUDENTE

Vuoi imparare a cucinare?

Esplora il vasto catalogo di corsi disponibili, iscriviti alle lezioni e inizia oggi stesso il tuo percorso culinario

[Accedi come Studente](#)

Pannello Login

Area Riservata Chef

Benvenuto, Giuseppe!

Gestisci i tuoi corsi e crea nuove esperienze culinarie

I Tuoi Corsi

[Filtro Corsi](#)

Preparazione torte	Inizio: 2025-05-12 - Frequenza: 2 volte a settimana
Pasticceria	Inizio: 2026-01-23 - Frequenza: martedì
Pasta e patate	Inizio: 2025-09-30 - Frequenza: Mensile
Fagioli e provola	Inizio: 2025-10-04 - Frequenza: Giornaliera

Gestione Corsi

Crea nuovi corsi, sessioni e ricette o gestisci quelli esistenti

[Crea nuovo corso](#)

[Gestisci Co...](#)

Pannello Area Riservata Chef

Crea Nuovo Corso

Compila tutti i campi per creare un nuovo corso culinario

Argomento del Corso

Es: Cucina Italiana Avanzata

Data Inizio

01 / 01 / 2025

Frequenza

Max Partecipanti

Es: 20

Modalità del Corso

Online In Presenza

Descrizione

Inserisci una descrizione dettagliata del corso...

Cancella
Crea Corso

Pannello Creazione di un nuovo corso

Gestione Corsi

[← Torna all'Area...](#)

Gestisci i tuoi corsi e comunica con gli studenti

Visualizza informazioni, iscritti e crea avvisi per i tuoi corsi

[I Tuoi Corsi](#)
[Filtro Corsi](#)

Preparazione torte

 Inizio: 2025-05-12 - Frequenza: 2 volte a settimana
 Iscritti: 3

[Azione](#)

Pasticceria

 Inizio: 2026-01-23 - Frequenza: martedì
 Iscritti: 7

[Azione](#)

Pasta e patate

 Inizio: 2025-09-30 - Frequenza: Mensile
 Iscritti: 2

[Azione](#)

Fagioli e provola

 Inizio: 2025-10-04 - Frequenza: Giornaliera
 Iscritti: 3

[Azione](#)

 Crea Nuovo Avviso X

Crea un Nuovo Avviso

Invia comunicazioni importanti ai tuoi studenti

Seleziona Corso

T234 - Preparazione torte

Titolo Avviso

Es: Cambio orario lezione, Materiali necessari...

Contenuto Avviso

Scrivi qui il messaggio completo per i tuoi studenti...

Annulla Crea Avviso

Dialog Creazione Avviso

Notifiche per Chef Giuseppe

Iscrizioni ai tuoi corsi e aggiornamenti importanti

← Torna all'Area ...

Nuova Iscrizione

✉ Nuova iscrizione al corso! Lo studente Federico Frisia si è iscritto al tuo corso: Preparazione torte

2025-09-30 19:03

Nuova Iscrizione

✉ Nuova iscrizione al corso! Lo studente Antonio Rossi si è iscritto al tuo corso: Preparazione torte

2025-09-30 19:01



Pannello Centro Notifiche

3

Corsi Totali

4

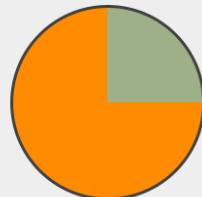
Sessioni Totali

1 Online | 3 Pratiche

Distribuzione Sessioni

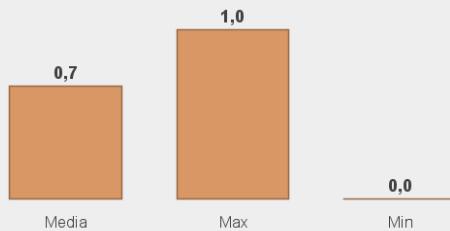
0,7 Media | 1 Max | 0 Min

Distribuzione Tipo Sessioni



■ Online (1) ■ Pratiche (3)

Statistiche Ricette (Sessioni Pratiche)



Pannello Report dello Chef

2.6 Class Diagram del Sistema

