

RicettApp

Bianca Bartoli - 0000874372

Lorenzo Medici - 0000881584

Francesco Olivo - 0000873790

Abstract.....	4
Analisi dei requisiti.....	5
Requisiti del sistema	5
Analisi del dominio.....	9
Dizionario	9
Applicazioni sul mercato	9
Analisi dei requisiti.....	10
Analisi del rischio	15
Tabella di valutazione dei beni	15
Tabella minacce/controlli	15
Analisi della tecnologia dal punto di vista della sicurezza	15
Descrizione delle interfacce grafiche	18
Analisi del problema	20
Analisi Documento dei Requisiti: Analisi delle Funzionalità	20
Tabella Funzionalità	20
Tabella informazioni/flusso.....	21
Analisi Documento dei Requisiti: Analisi dei Vincoli	24
Tabella dei vincoli.....	24
Analisi Documento dei Requisiti: Analisi delle Interazioni.....	24
Tabella delle maschere	24
Analisi Ruoli e Responsabilità	25
Tabella dei ruoli.....	25
Tabella Ruolo-Informazioni	25
Scomposizione del problema.....	25
Tabella scomposizione funzionalità	25
Analisi delle sottofunzionalità.....	26
Modello del dominio.....	35
Architettura logica	36
Struttura.....	36
Interazione	40
Comportamento.....	42
Piano di lavoro	43
Sviluppi futuri.....	43
Piano di collaudo.....	44

Progettazione.....	46
Progettazione architetturale.....	46
Requisiti non funzionali.....	46
Scelta dell'architettura.....	46
Scelte tecnologiche	48
Progettazione di dettaglio.....	49
Struttura.....	49
Interazione	57
Comportamento.....	58
Progettazione persistenza	59
Progettazione del collaudo	60
Deployment	61
Artefatti.....	61
Deployment type-level.....	61

Abstract

La pandemia negli ultimi mesi ha comportato la chiusura dei ristoranti e la necessità di recarsi poche volte a fare la spesa, richiedendo così una buona organizzazione alimentare alle famiglie che si sono trovate più che mai impegnate ai fornelli e a sperimentare nuovi piatti.

Poiché negli ultimi mesi ci sono giunte numerose richieste da parte delle nostre famiglie che ci chiedevano di pianificare i pasti dei giorni successivi, abbiamo pensato di progettare un'applicazione che permetta, oltre alle classiche funzionalità di un ricettario, di pianificare e organizzare in modo automatico le ricette da preparare, permettendo una forte personalizzazione data l'estrema varietà delle esigenze alimentari di oggi.

Analisi dei requisiti

Requisiti del sistema

L'idea è di creare un ricettario che permetta una gestione avanzata di un ricettario: l'utente può aggiungere una ricetta in vari modi, che devono essere estendibili e modificabili.

Ogni ricetta ha delle caratteristiche, ad esempio portata, complicatezza, lista di ingredienti (quindi allergie o intolleranze) legate alla quantità di persone, tempistiche, apporto calorico. Una ricetta può presentare una o più foto, una lista di ingredienti e una sequenza di passaggi.

Un utente può iniziare a svolgere una o più ricette in contemporanea. Durante lo svolgimento l'utente vede solo il passaggio che deve fare nell'immediato, pur potendo scorrere comodamente tra i passaggi; un pulsante apposito deve permettere di poter visualizzare sempre la lista degli ingredienti.

Ogni ingrediente presenta delle caratteristiche quali calorie, stagionalità, l'essere o meno un allergene.

Un utente può aggiungere una propria ricetta attraverso l'apposita interfaccia al proprio ricettario.

Al primo avvio l'utente ha sull'app una piccola lista di ricette, di tag e di ingredienti di default.

Un utente può aggiungere un ingrediente. Non è obbligatorio inserire i valori nutrizionali e la stagionalità.

Un utente quando inizia una ricetta può selezionare il numero di persone e il ricettario calcolerà in automatico le dosi di ingredienti.

Ogni utente dispone di un "frigo" e di una lista della spesa. Il frigo contiene tutti gli ingredienti a disposizione dell'utente, la lista della spesa tutti gli ingredienti che l'utente vuole acquistare. Gli ingredienti spuntati dalla lista della spesa vengono automaticamente aggiunti al frigo. Un utente potrà sempre modificare manualmente tutto.

Un utente può cercare una ricetta basandosi su diversi fattori: fondamentale è garantire una forte personalizzazione dei fattori di ricerca: alcuni esempi sono cercare ricette che contengano solo ingredienti presenti in frigo o in frigo e nella lista della spesa, ricette che non contengano determinati allergeni, ricette con un certo apporto calorico, stagionalità, eccetera.

Quando un utente termina l'esecuzione di una ricetta gli viene chiesto se ha terminato qualche ingrediente della ricetta, e nel caso se vuole aggiungerli alla lista della spesa.

Un utente può creare uno o più tag da aggiungere a una ricetta, tag che potrà poi selezionare in fase di ricerca.

Un utente può registrarsi all'applicazione. Un utente registrato ha le sue ricette e tutti i suoi dati sul cloud e può trasferirli su più dispositivi.

In uno scenario più realistico un utente potrebbe avere più di una lista della spesa, per semplicità noi permettiamo che ne possa avere una sola.

In uno scenario reale un utente potrebbe voler salvare una lista di allergeni che non devono comparire a prescindere (o particolari diete) nella ricerca. Questo requisito esula dallo scopo del nostro progetto.

È disponibile un calendario, dove ogni giorno è diviso nei pasti principali: l'utente può aggiungere a ogni pasto una delle ricette presenti nel suo ricettario. È inoltre presente una sezione per raccogliere casi particolari e vari (ad esempio una torta)

Se l'utente dal calendario clicca su una ricetta si apre la visualizzazione della ricetta.

Un utente può aggiungere una ricetta al calendario in due modi, o dal calendario, nel tal caso si apre il menù di ricerca, o direttamente da una ricetta.

Un utente può rimuovere quando vuole una ricetta dal calendario.

Un utente può effettuare una ricerca tra gli ingredienti usando come parametri nome, stagionalità e tipologia. La ricerca deve essere estendibile in vista di estensioni future.

Un utente può registrarsi al sistema.

Un utente registrato ha automaticamente le sue ricette sul cloud e nel momento in cui esegue l'accesso da un altro dispositivo le ricette vengono automaticamente scaricate in locale.

Legenda:

1. RU: utente
2. RS: ricerca (search)
3. RR: ricetta
4. RI: ingrediente
5. RF: frigo
6. RL: list
7. RE: esecuzione ricetta
8. RT: tag
9. RC: calendario
10. RD: database

ID REQUISITO	REQUISITO	TIPO
R1EF	Un utente può eseguire contemporaneamente un numero finito di ricette.	Requisito funzionale
R2EF	Durante l'esecuzione un utente visualizza un passaggio alla volta, e può scorrere tra i passaggi.	Requisito funzionale
R3EF	Durante l'esecuzione un utente può visualizzare in ogni momento la lista degli ingredienti.	Requisito funzionale
R4EF	Quando un utente inizia l'esecuzione di una ricetta, può selezionare il numero di persone per cui cucina.	Requisito funzionale
R5EF	Il primo passaggio di ogni ricetta è la lista di ingredienti.	Requisito funzionale
R6EF	Al termine dell'esecuzione viene chiesto all'utente se ha finito qualche ingrediente, e nel caso se vuole aggiungerlo alla lista della spesa.	Requisito funzionale
R7ENF	Il numero massimo di ricette in esecuzione dipende dallo schermo a disposizione dell'utente.	Requisito non funzionale

R1FD	Il frigorifero contiene un insieme di ingredienti.	Requisito di dominio
R2FF	L'utente può aggiungere, rimuovere o modificare ogni ingrediente nel frigo.	Requisito funzionale
R1ID	Di ogni ingrediente si registrano tipologia, calorie e stagionalità, unità di misura.	Requisito di dominio
R2IF	L'utente può visualizzare la lista degli ingredienti e eseguire una ricerca per nome, eventualmente estendibile in futuro	Requisito funzionale
R3IF	L'utente può aggiungere ingredienti alla lista degli ingredienti o modificare ingredienti già presenti	Requisito funzionale
R1LD	Una lista della spesa contiene un insieme di coppie ingrediente e quantità opzionale.	Requisito di dominio
R2LF	L'utente può aggiungere, rimuovere, modificare o comprare ogni ingrediente nella lista della spesa.	Requisito funzionale
R3LF	Quando un utente compra un ingrediente dalla lista della spesa questo viene automaticamente aggiunto al frigorifero e rimosso dalla lista.	Requisito funzionale
R4LF	Gli ingredienti aggiunti automaticamente alla lista si presentano senza quantità e sarà l'utente a decidere se rimuoverli o se acquistarli	Requisito funzionale
R1RD	Di ogni ricetta si registrano un insieme di tag, una lista di ingredienti, il grado di difficoltà, una lista di passaggi, un tempo di esecuzione, l'apporto calorico e opzionalmente una foto	Requisito di dominio
R2RD	Ogni ricetta associa a ogni ingrediente una dose opzionale	Requisito di dominio
R3RF	Quando un utente visita una ricetta può verificare le dosi di ingredienti in base al numero di persone.	Requisito funzionale
R1SF	Un utente può cercare delle ricette dal ricettario utilizzando tag, ingredienti che ha nel frigo e/o nella lista della spesa e rimuovendo ingredienti a cui è allergico o che non gli piacciono, tempo di esecuzione o apporto calorico.	Requisito funzionale
R2SF	Un utente può cercare una ricetta per nome, combinandolo o meno con i requisiti del parametro sopra.	Requisito funzionale
R3SNF	La ricerca non deve avere tempi di caricamento eccessivi	Requisito non funzionale
R1TD	Un tag presenta un nome.	Requisito di dominio
R2TNF	Ogni utente può creare un numero finito molto alto di tag.	Requisito non funzionale
R3TF	L'utente può modificare o rimuovere qualsiasi tag	Requisito funzionale
R1UNF	Ogni utente dispone di un ricettario.	Requisito non funzionale
R2UNF	Ogni utente dispone di una lista degli ingredienti	Requisito non funzionale
R3UNF	Ogni utente dispone di un frigo	Requisito non funzionale
R4UNF	Ogni utente dispone di una lista della spesa	Requisito non funzionale
R5UNF	Ogni utente dispone di un insieme di tag.	Requisito non funzionale

R6UF	Ogni utente può aggiungere al ricettario un numero finito molto alto di nuove ricette, utilizzando in futuro svariati metodi di persistenza.	Requisito funzionale
R7UF	Ogni utente può modificare liberamente ogni ricetta appartenente al ricettario.	Requisito funzionale
R8UNF	Ogni utente dispone di un calendario	Requisito non funzionale
R1CD	Un calendario è diviso in giorni	Requisito di dominio
R2CD	Ogni giorno è organizzato in pasti.	Requisito di dominio
R3CF	Un utente può aggiungere a un pasto una ricetta da preparare, sia dall'apposita vista sul calendario che dalla pagina di una ricetta.	Requisito funzionale
R4CF	L'utente può rimuovere una ricetta assegnata ad un pasto	Requisito funzionale
R1DF	L'utente può registrarsi.	Requisito funzionale
R2DF	Un utente registrato può eseguire l'accesso o disconnettersi dall'applicazione.	Requisito funzionale
R3DF	Tutti i dati dell'utente vengono caricati su un server al momento della registrazione.	Requisito funzionale
R4DF	Ogni volta che un utente registrato e che ha eseguito l'accesso aggiunge/modifica/elimina un dato all'interno dell'applicazione, la modifica viene trasmessa al server.	Requisito funzionale
R5DF	Quando un utente già registrato esegue l'accesso da un nuovo dispositivo tutti i suoi dati presenti sul server vengono scaricati.	Requisito funzionale
R6DF	All'avvio dell'applicazione si consulta il server per vedere se sono presenti nuovi dati, e in caso affermativo questi vengono caricati anche in locale.	Requisito funzionale
R1NF	L'applicazione deve essere facilmente estendibile, visti gli sviluppi futuri già pianificati	Requisito non funzionale
R2NF	Le interfacce utente devono essere semplici e intuitive, adatte a un pubblico il più ampio possibile.	Requisito non funzionale
R3NF	L'applicazione non deve essere troppo esosa in termini di risorse hardware.	Requisito non funzionale
R4NF	L'applicazione non deve avere tempi di caricamento che siano eccessivamente lunghi per un utente (massimo pochi secondi)	Requisito non funzionale

Analisi del dominio

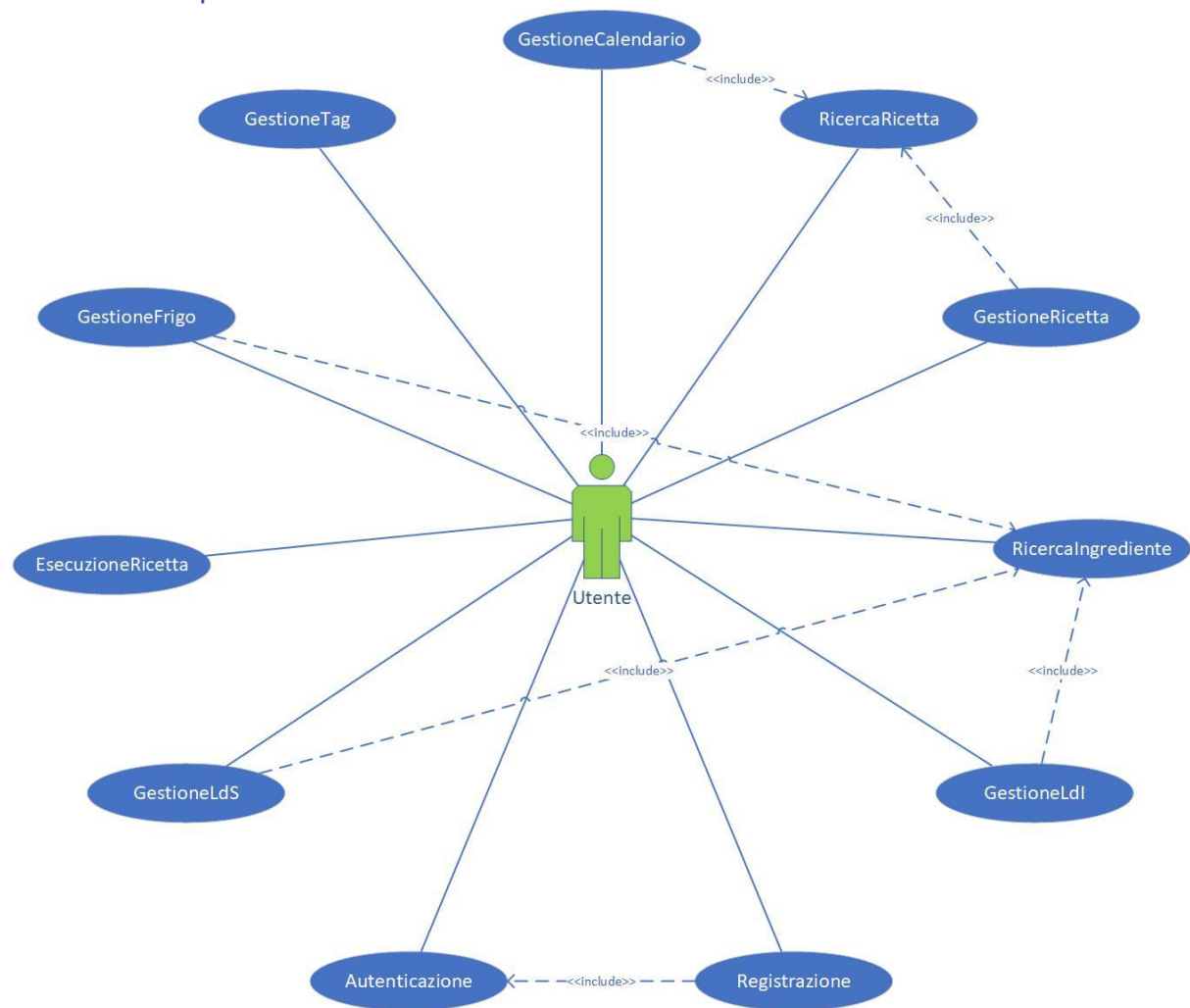
Dizionario

- **Acquisto:** passaggio in cui un utente segna come realmente acquistato un ingrediente presente nella lista della spesa.
- **Apporto calorico di un ingrediente:** quantità di calorie per unità di misura che contiene un ingrediente.
- **Apporto calorico di una ricetta:** Quantità di calorie del piatto ottenuto dalla ricetta, calcolata come somma degli apporti calorici di ogni ingrediente per quantità diviso il numero delle persone.
- **Calendario dei pasti:** insieme dei pasti pianificati dall'utente
- **Dose di un ingrediente:** quantità opzionale di un ingrediente utilizzato in una ricetta.
- **Esecuzione:** la sequenza di gesti che l'utente compie per trasformare gli ingredienti in un piatto finito.
- **Frigorifero:** metafora virtuale del frigorifero e della dispensa di un utente: idealmente contiene tutti gli ingredienti di cui dispone sul momento l'utente. Sinonimi: **Frigo, dispensa**.
- **Ingrediente:** qualunque alimento l'utente utilizzi per realizzare un piatto.
- **Lista della spesa:** insieme di ingredienti che l'utente si prefissa o ha intenzione di comprare in un periodo successivo.
- **Numero di persone:** caratteristica di una ricetta che indica per quante persone sono calcolate le dosi.
- **Passaggio:** Istruzione specifica che l'utente esegue per trasformare uno o più ingredienti.
- **Pasto:** insieme di ricette che l'utente si prefissa di preparare in un tempo ravvicinato.
- **Pianificazione di una ricetta:** azione con cui l'utente assegna una ricetta a un pasto di uno specifico giorno
- **Ricetta:** Insieme di passaggi, ingredienti e informazioni che l'utente segue per cucinare un piatto.
- **Ricettario:** insieme di ricette: una ricetta può comparire più volte nel ricettario ma con un nome diverso, come nel caso di versioni diverse di una stessa ricetta.
- **Stagionalità di un ingrediente:** periodo dell'anno in cui l'alimento è disponibile; si applica soprattutto ad alimenti freschi quali frutta, verdura e pesce.
- **Tag:** etichetta che viene aggiunta a una o più ricette per determinare caratteristiche e proprietà.
- **Tempo di esecuzione:** quantità di tempo medio richiesta per svolgere l'insieme di passaggi di una ricetta.
- **Terminazione di un ingrediente:** quando la quantità disponibile di un ingrediente nel frigorifero diventa nulla o scende sotto una dose determinata dall'utente.
- **Tipologia di un ingrediente:** a quale famiglia di alimenti l'ingrediente appartiene (esempio: frutta, verdura, pesce, carne)
- **Unità di misura di un ingrediente:** metodologia di misurazione della quantità di un ingrediente.
- **Utente:** qualunque persona che utilizza l'applicazione

Applicazioni sul mercato

La principale applicazione simile sul mercato attualmente è Cookbook, le cui funzionalità sono tutto sommato simili a quelle di Ricettapp, per quanto ne siamo giunti a conoscenza solo dopo l'analisi dei requisiti. Quello che differenzia particolarmente la nostra applicazione è la gestione degli ingredienti e del frigo, che permette all'utente di avere maggiore consapevolezza verso gli sprechi alimentari e all'apporto calorico di quanto mangia.

Analisi dei requisiti



RicercaRicetta

Descrizione	Ricerca di una ricetta
Attori	Utente
Relazioni	
Precondizioni	L'utente ha un insieme di ricette
Postcondizioni	Trova la ricetta desiderata e nel caso la esegue
Scenari principali	<ol style="list-style-type: none"> 1. L'utente effettua una ricerca per tag o per ingredienti voluti o non voluti, presenti in frigo e/o nella lista della spesa (eventualmente anche per calorie) 2. ottiene una lista di ricette che soddisfano la ricerca, da cui poi potrà scegliere quella da visualizzare a schermo intero
Scenari alternativi	
Requisiti non funzionali	La ricerca deve essere rapida
Punti aperti	Cosa succede se non trovo nessuna ricetta? Fino a quanti filtri può applicare un utente per fare una ricerca

RicercaIngrediente

Descrizione	Ricerca di un ingrediente
Attori	Utente
Relazioni	
Precondizioni	L'utente ha una lista di ingredienti (di default + aggiunti personalmente)
Postcondizioni	Ottiene la scheda dell'ingrediente cercato
Scenari principali	<ol style="list-style-type: none">1. L'utente effettua una ricerca sul nome dell'ingrediente desiderato2. ottiene la sua scheda che può consultare per vedere le sue proprietà
Scenari alternativi	
Requisiti non funzionali	La ricerca deve essere rapida
Punti aperti	Se un ingrediente non è presente? L'utente può aggiungerlo? Se l'ingrediente è salvato con un altro nome?

EsecuzioneRicetta

Descrizione	Esecuzione di una ricetta
Attori	Utente
Relazioni	Terminazione di un ingrediente (<i>include</i>)
Precondizioni	Esiste almeno una ricetta.
Postcondizioni	Nessuna
Scenari principali	Dal menù visualizzazione <ol style="list-style-type: none">1. l'utente avvia l'esecuzione della ricetta.2. visualizzare gli ingredienti mediante un bottone o scorrere tra i passaggi.3. segnala la fine della ricetta4. gli viene chiesto se ha terminato qualche ingrediente e vuole aggiungerlo alla lista della spesa.
Scenari alternativi	
Requisiti non funzionali	I comandi devono essere intuitivi e non diventare troppo piccoli
Punti aperti	

GestioneLdS

Descrizione	Gestione della lista della spesa
Attori	Utente
Relazioni	AggiungiIngredienteLdS, ModificaIngredienteLdS, AcquistoIngredienteLdS, EliminaIngredienteLdS
Precondizioni	
Postcondizioni	
Scenari principali	<ol style="list-style-type: none">1. L'utente visualizza gli ingredienti presenti nella lista della spesa2. L'utente può aggiungere un ingrediente toccando un apposito pulsante3. L'utente può modificare la quantità da acquistare di un ingrediente, segnalarlo come acquistato o rimuoverlo toccando gli appositi pulsanti
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

GestioneRicetta

Descrizione	Gestione delle ricette
Attori	Utente
Relazioni	AggiungiRicetta, ModificaRicetta, EliminaRicetta
Precondizioni	
Postcondizioni	
Scenari principali	<ol style="list-style-type: none">1. L'utente visualizza le ricette esistenti2. L'utente può aggiungere una ricetta toccando un apposito pulsante3. Una volta visualizzata una ricetta l'utente può modificarla o eliminarla tramite appositi pulsanti
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

GestioneFrigo

Descrizione	Gestione del frigo
Attori	Utente
Relazioni	AggiungiIngredienteFrigo, TerminazioneIngredienteFrigo, ModificaIngredienteFrigo, EliminaIngredienteFrigo
Precondizioni	
Postcondizioni	
Scenari principali	<ol style="list-style-type: none">1. L'utente visualizza la lista degli ingredienti presenti nel frigo2. L'utente può aggiungerne uno tramite l'apposito pulsante3. L'utente può dichiararne uno come terminato o rimuoverlo tramite gli appositi pulsanti associati a quell'ingrediente
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

GestioneLdl

Descrizione	Gestione della lista degli ingredienti
Attori	Utente
Relazioni	Aggiunta di un ingrediente, rimozione di un ingrediente, modifica di un ingrediente
Precondizioni	
Postcondizioni	
Scenari principali	<ol style="list-style-type: none">1. L'utente visualizza gli ingredienti esistenti2. L'utente può aggiungere un ingrediente toccando un apposito pulsante3. L'utente può modificare o rimuovere un ingrediente toccandone il pulsante associato
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

GestioneCalendario

Descrizione	Gestione del calendario
Attori	Utente
Relazioni	AggiungiRicettaCalendario, EliminaRicettaCalendario
Precondizioni	
Postcondizioni	
Scenari principali	<ol style="list-style-type: none">1. L'utente visualizza il calendario e può scegliere un giorno per vedere i pasti assegnati e non di quel giorno2. L'utente assegnare una ricetta ad un pasto libero toccandolo3. L'utente può rimuovere una ricetta dal pasto a cui è stata assegnata
Scenari alternativi	Dalla visualizzazione della ricetta l'utente può toccare un pulsante e scegliere giorno e pasto a cui assegnare la ricetta visualizzata
Requisiti non funzionali	
Punti aperti	

GestioneTag

Descrizione	Gestione dei tag
Attori	Utente
Relazioni	AggiungiTag, ModificaTag, EliminaTag
Precondizioni	
Postcondizioni	
Scenari principali	<ol style="list-style-type: none">1. L'utente visualizza i tag esistenti2. L'utente può aggiungere un tag toccando un apposito pulsante3. L'utente può modificare un tag toccandone il pulsante associato
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

Autenticazione

Descrizione	Autenticazione
Attori	Utente
Relazioni	Registrazione
Precondizioni	L'utente è registrato
Postcondizioni	L'utente ha fatto l'accesso
Scenari principali	<ol style="list-style-type: none">1. L'utente inserisce nome utente e password2. Se le credenziali sono corrette l'utente ha effettuato l'accesso, che verrà effettuato in automatico ad ogni avvio dell'applicazione
Scenari alternativi	Scenario a: Le credenziali sono errate <ol style="list-style-type: none">1. L'applicazione comunica all'utente che le credenziali sono errate2. L'utente può ritentare l'accesso
Requisiti non funzionali	Sicurezza della comunicazione col server, sicurezza della memorizzazione locale
Punti aperti	

Registrazione

Descrizione	Registrazione
Attori	Utente
Relazioni	Autenticazione (<i>include</i>)
Precondizioni	L'utente non è registrato
Postcondizioni	L'utente è registrato e ha automaticamente effettuato l'accesso
Scenari principali	<ol style="list-style-type: none">1. L'utente inserisce username, password ed e-mail2. L'applicazione conferma la registrazione ed effettua l'accesso in automatico
Scenari alternativi	<p>Scenario a: Il nome utente o la e-mail sono già registrati</p> <ol style="list-style-type: none">1. L'applicazione comunica all'utente quale dei campi non è valido2. L'utente può inserire nuovi dati per ritentare l'autenticazione <p>Scenario b: Password troppo corta</p> <ol style="list-style-type: none">1. L'applicazione comunica all'utente che la password è troppo breve2. L'utente può inserire una nuova password per tentare la registrazione
Requisiti non funzionali	Sicurezza della comunicazione col server
Punti aperti	

Analisi del rischio

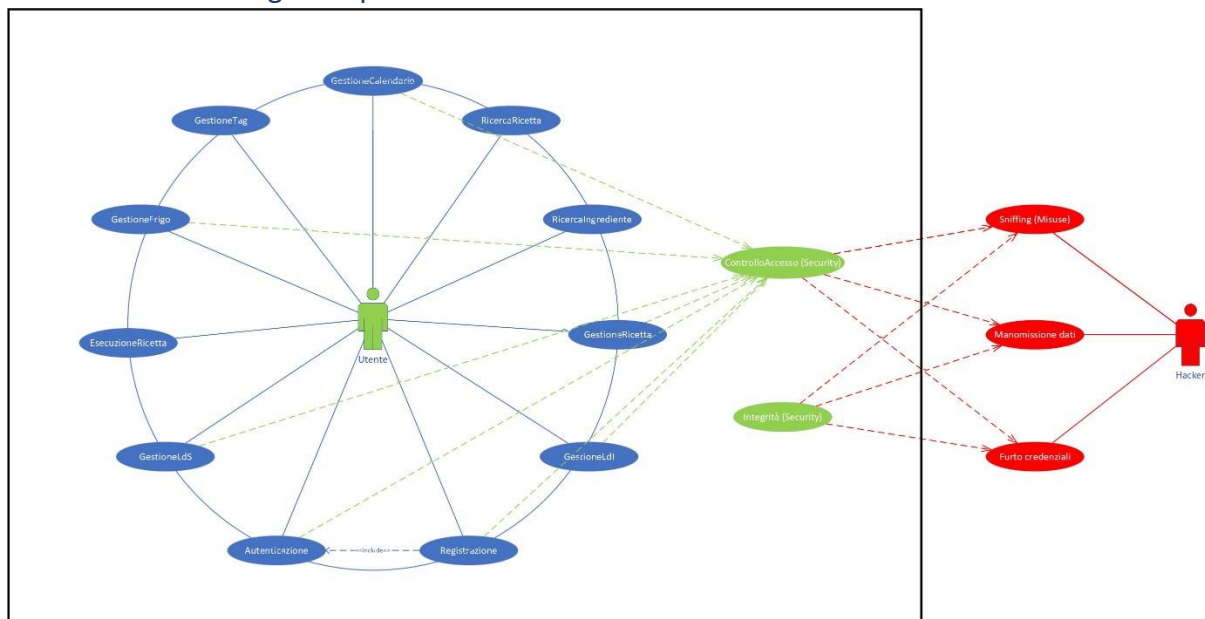
Tabella di valutazione dei beni

Bene	Valore	Esposizione
Informazioni relative agli Utenti	Alto. Sono presenti informazioni sensibili quali mail e password.	Alta. Danni di immagine in caso di divulgazione dei dati.
Dati applicativi dell'utente	Basso. Ricette e ingredienti usati non sono dati particolarmente sensibili.	Bassa. Difficile immaginare uno scenario in cui questi dati verranno usati contro l'utente.

Tabella minacce/controlli

Minaccia	Probabilità	Controllo	Fattibilità
Furto credenziali Utente	Alta	Log delle operazioni.	Basso costo implementativo.
Manomissione dei dati applicativi dell'utente.	Bassa	Log delle operazioni	Basso costo.
Intercettazione delle comunicazioni	Media. Il sistema sfrutta il modello client/server quindi ci sono molte interazioni tra i client e il server.	Cifratura delle comunicazioni.	Variabile in base al tipo di cifratura scelto. Si sceglie una cifratura simmetrica con basso costo implementativo, data la non particolare importanza dei dati scambiati.
Denial of Service	Media	Progettazione adeguata, controllo e limitazione degli accessi.	Basso costo. Impossibile prevedere attacchi di questo tipo.

Analisi della tecnologia dal punto di vista della sicurezza



Tecnologia	Vulnerabilità
Architettura Client/Server	<ul style="list-style-type: none"> • Attacco Denial of Service • Attacco Man in the Middle • Intercettazione delle comunicazioni
Cifratura delle informazioni	<p>A seconda del tipo di cifratura utilizzata si hanno diverse vulnerabilità:</p> <ol style="list-style-type: none"> 1. Cifratura Asimmetrica <ul style="list-style-type: none"> - Affidabilità e memorizzazione della chiave privata 2. Cifratura Simmetrica <ul style="list-style-type: none"> - Tempo di vita della chiave - Lunghezza della chiave - Memorizzazione della chiave
Autenticazione tramite username/password	<ul style="list-style-type: none"> • Password semplici • Poca sicurezza da parte dell'utente • Password semplici

Titolo	ControlloAccesso	
Descrizione	Ogni accesso al server deve essere controllato	
Misuse Case	FurtoCredenziali, ManomissioneRicettario, SniffingInformazioni	
Relazioni		
Precondizioni	L'attaccante è in grado di scoprire username e password degli utenti	
Postcondizioni	Il server blocca temporaneamente l'accesso all'utente e propone la modifica della password via e-mail	
Scenario principale	Sistema	Attaccante
		L'attaccante tenta l'accesso con username e password di cui non conosce la validità
	Il server controlla le credenziali immesse e scrive nel log il tentativo di accesso. Al terzo tentativo errato l'utente viene bloccato temporaneamente e gli viene inviata una e-mail per scegliere una nuova password	
Scenario di attacco avvenuto con successo	Sistema	Attaccante
		L'attaccante tenta l'accesso con credenziali valide
	Il server controlla le credenziali e consente l'accesso ai dati salvati	
		L'attaccante può scaricare i dati dell'utente legittimo e salvare sul server dati modificati
	Il server scrive sul log le operazioni eseguite dall'utente	

Titolo	Integrità	
Descrizione	Integrità dei dati salvati dal server	
Misuse Case	FurtoCredenziali, ManomissioneRicettario, SniffingInformazioni	
Relazioni		
Precondizioni	Il server ha memorizzato i dati dell'utente, che devono essere acceduti solo dall'utente stesso	
Postcondizioni	Il server registra tutte le operazioni effettuate	
Scenario principale	Sistema	Attaccante
		L'attaccante cerca di corrompere i dati
	Il server impedisce qualsiasi modifica dei dati non proveniente da utenti che hanno effettuato l'accesso	
Scenario di attacco avvenuto con successo	Sistema	Attaccante
		L'attaccante riesce a corrompere i dati dell'utente
	Si usano i dati presenti in locale sul client	

Descrizione delle interfacce grafiche



Figura 2: visualizzazione del ricettario

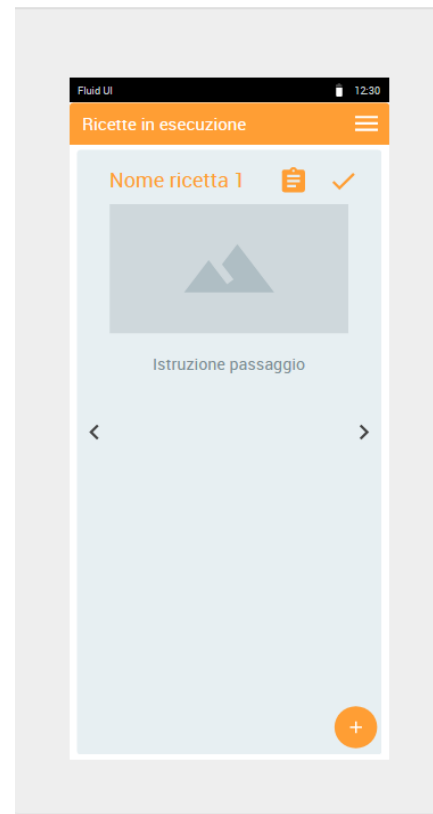


Figura 1: esecuzione di una ricetta

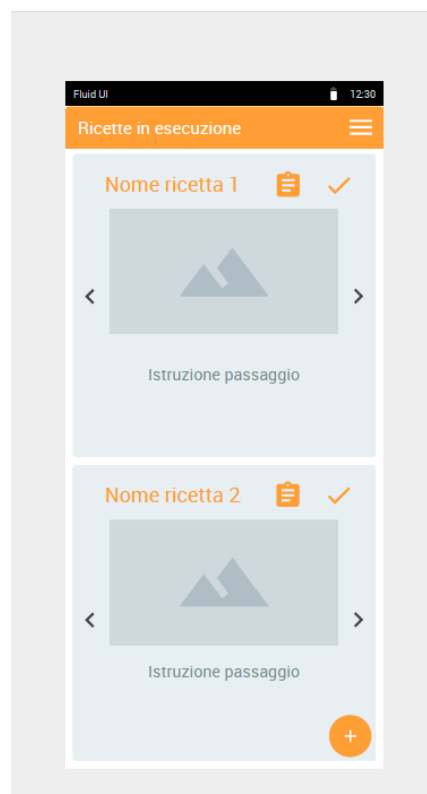


Figura 3: esecuzione di più ricette

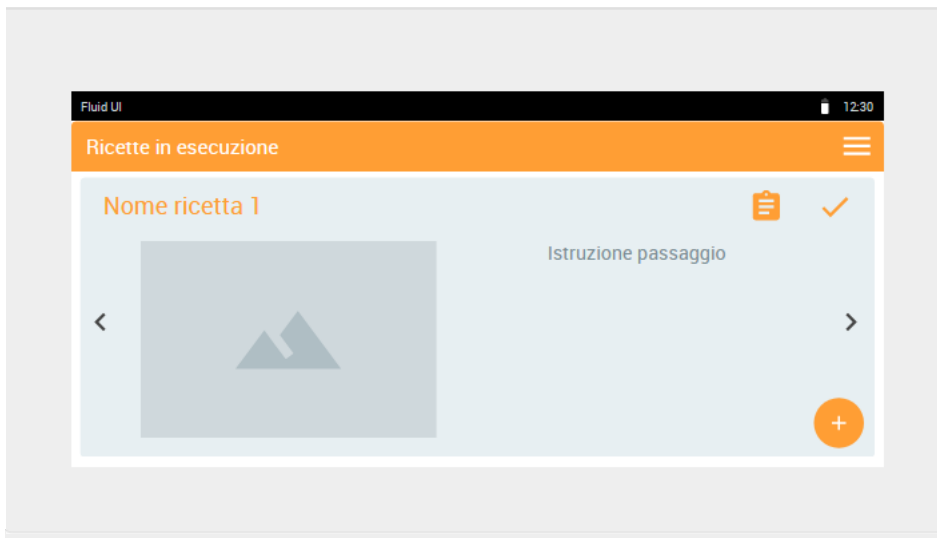


Figura 4: esecuzione di una ricetta con il dispositivo in orizzontale

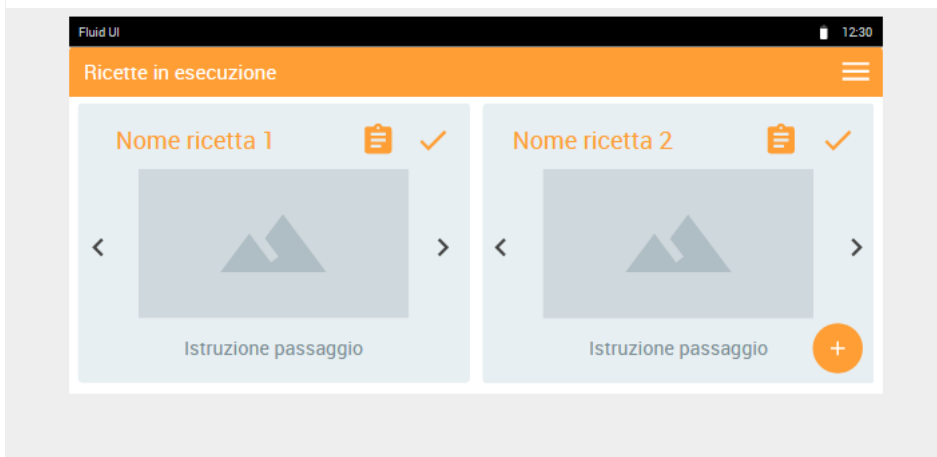


Figura 5: esecuzione di più ricette con il dispositivo in orizzontale

Prototipi realizzati con FluidUI.

Analisi del problema

Analisi Documento dei Requisiti: Analisi delle Funzionalità

Tabella Funzionalità

Funzionalità	Tipo	Grado complessità	Requisiti Collegati
Gestione Ricetta	Gestione e memorizzazione dati	Complessa	R1RD, R6UF, R7UF
Ricerca Ingrediente	Interazione con l'utente, gestione dati	Semplice	R2SF, R2IF, R3SNF
Gestione Frigo	Gestione e memorizzazione dati	Complessa	R1FF, R2FF
Gestione Lista della Spesa	Gestione e memorizzazione dati	Complessa	R1LD, R2LF, R4LF
Acquisto Ingrediente	Gestione e memorizzazione dati	Semplice	R3LF
Esecuzione Ricetta	Interazione con l'utente	Semplice	R1EF, R2EF, R3EF, R4EF, R5EF, R7ENF
Terminazione Ingrediente	Gestione e memorizzazione dati	Semplice	R6EF
Gestione Lista degli Ingredienti	Gestione e memorizzazione dati	Complessa	R2FI
Gestione Calendario	Gestione e memorizzazione dati	Complessa	R1CF, R2CF, R3CF, R4CF
Ricerca Ricetta	Interazione con l'utente, gestione dati	Semplice	R1SF, R3SNF
Gestione Tag	Gestione e memorizzazione dati	Complessa	R1TD, R2TNF, R3TF
Autenticazione	Interazione con l'utente, controllo dell'accesso	Semplice	R2DF, R4DF, R5DF, R6DF,
Registrazione	Interazione con l'utente, controllo dell'accesso	Semplice	R1DF, R3DF

Tabelle informazioni/flusso

GestioneLdl: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
ListaIngredienti	Composta	Bassa	Input/Output	

AggiungiIngredienteLdl: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
Ingrediente composto da	Composto	Bassa	Output	
Nome	Semplice	Bassa	Input	
Tipologia	Semplice	Bassa	Input	Enumerativo
Calorie composto da	Composta	Bassa	Output	
Quantità calorie	Semplice	Bassa	Input	Bassa
Denominatore	Semplice	Bassa	Input	Valore numerico
Stagionalità composto da	Composta	Bassa	Output	
Inizio	Semplice	Bassa	Input	Intero tra 1 e 12
Fine	Semplice	Bassa	Input	Intero tra 1 e 12
ListaFotografie composta da	Composta	Bassa	Output	
Fotografia	Semplice	Bassa	Input	File binario con estensione jpg o png

ModificaIngredienteLdl: vedi aggiunta ingrediente

EsecuzioneRicetta: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
--------------	------	--------------	--------------	---------

TerminazioneIngrediente: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
Ingrediente	Composto	Bassa	Input	
Quantità	Semplice	Bassa	Input	Opzionale
Unità di misura	Semplice	Bassa	Input	Opzionale, enumerativo

GestioneCalendario: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
Pasto composto da	Composto	Bassa	Output	
Data	Semplice	Bassa	Input	
Tipologia	Semplice	Bassa	Input	Enumerativo es. Pranzo cena etc.
ListaRicette composto da	Composto	Bassa	Input	
Ricetta	Composto	Bassa	Input	

RicercaRicetta: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
ListaRicette composto da	Composto	Bassa	Output	
Ricetta	Composto	Bassa	Output	
Nome	Semplice	Bassa	Input	
ListaIngredientiOK	Composta	Bassa	Input	
ListaIngredientiNo	Composta	Bassa	Input	
ListaTag	Composta	Bassa	Input	

GestioneTag: Tabella Informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
ListaTag composto da	Composto	Bassa	Output	
Tag composto da	Composto	Bassa	Output	
Nome	Semplice	Bassa	Input	

GestioneFrigo: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
ListaIngredienti	Composta	Bassa	Input/Output	

GestioneRicetta: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
Nome	Semplice	Bassa	Input	Almeno 32 caratteri
Timestamp	Semplice	Bassa	Input/Output	LocalDateTime
ListaTag composto da	Composta	Bassa	Input/Output	Limitato a 100
Tag	Semplice	Bassa	Input/Output	Almeno 32 caratteri
ListaIngredienti	Composta	Bassa	Input/Output	
Foto	Semplice	Bassa	Input	
ListaPassaggi	Composta	Bassa	Input/Output	

RicercaIngrediente: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
Nome	Semplice	Bassa	Input/Output	Almeno 32 caratteri
ListaTag composto da	Composta	Bassa	Input/Output	Limitato a 100
Tag	Semplice	Bassa	Input/Output	Almeno 32 caratteri
Calorie composto da	Composta	Bassa	Output	
Quantità cal	Semplice	Bassa	Input	Valore Numerico
Denominatore	Semplice	Bassa	Input	Valore Numerico
Stagionalità composto da	Composta	Bassa	Output	
Inizio	Semplice	Bassa	Input	Intero tra 1 e 12
Fine	Semplice	Bassa	Input	Intero tra 1 e 12

GestioneIngredienteLds: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
Ingrediente composto da	Composto	Bassa	Output	
Nome	Semplice	Bassa	Input	
Tipologia	Semplice	Bassa	Input	Enumerativo
Calorie composto da	Composta	Bassa	Output	
Quantità cal	Semplice	Bassa	Input	Bassa
Denominatore	Semplice	Bassa	Input	Valore numerico
Stagionalità composto da	Composta	Bassa	Output	
Inizio	Semplice	Bassa	Input	Intero tra 1 e 12
Fine	Semplice	Bassa	Input	Intero tra 1 e 12
ListaFotografie composta da	Composta	Bassa	Output	
Fotografia	Semplice	Bassa	Input	File binario con estensione jpg o png

AcquistiIngredienteLds: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
ListaIngredienti Frigo	Composta	Bassa	Output	
Ingrediente Lds	Semplice	Bassa	Input	

Autenticazione: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
Nome utente	Semplice	Alta	Input	
Password	Semplice	Alta	Input	

Registrazione: tabella informazioni/flusso

Informazione	Tipo	Riservatezza	Input/Output	Vincoli
Nome utente	Semplice	Alta	Input	
Password	Semplice	Alta	Input	
Indirizzo e-mail	Semplice	Alta	Input	

Analisi Documento dei Requisiti: Analisi dei Vincoli

Tabella dei vincoli

Requisito	Categorie	Impatto	Funzionalità
Facilità di navigazione della UI, pulsanti essenziali e chiari	Usabilità	Rende l'applicativo più semplice e a portata dell'utente	Tutta la parte client
Velocità di reazione e ridotti tempi di caricamento	Usabilità	Richiede progettazione specifica	GestioneIngredienti, GestioneRicettario, GestioneCalendario
Protezione dei dati	Sicurezza	Rallenta il sistema per garantire privacy dei dati.	Tutta parte client

Analisi Documento dei Requisiti: Analisi delle Interazioni

Tabella delle maschere

Maschera	Informazioni	Funzionalità
View Ricettario	Ricette disponibili o cercate	Ricerca Ricetta, Aggiungi Ricetta
View Ricerca Ricetta	Parametri di ricerca	Ricerca Ricetta
View Aggiunta ricetta	Attributi di una ricetta	Aggiungi Ricetta, Modifica Ricetta, AggiungiIngrediente, AggiungiTag
View Ricetta	Dati di una ricetta	AggiuntaCalendario, ModificaRicetta, EliminaRicetta, EsecuzioneRicetta
View Esecuzione	Passaggi di una o più ricette	EsecuzioneRicetta, TerminazioneIngrediente
View Lista Ingredienti	Ingredienti disponibili o cercati	Ricerca Ingrediente, Aggiungi Ingrediente
View Ingrediente	Attributi di un ingrediente	ModificaIngrediente
View Frigo	Lista di ingredienti nel frigo	Aggiungi Ingrediente Frigo, Rimuovi Ingrediente Frigo, Terminazione Ingrediente
View Ricerca Ingrediente	Parametri di ricerca	Ricerca Ingrediente
View Aggiunta Ingrediente	Attributi di un ingrediente	Aggiungi Ingrediente, Modifica Ingrediente
View Lista della spesa	Ingredienti nella lista della spesa	AggiungiIngredienteListaDellaSpesa, EliminaIngredienteListaDellaSpesa, AcquistaIngrediente
View Calendario	Pasti organizzati	AggiungiPastoAlCalendario
View lista di tag	Tag	AggiungiTag, ModificaTag, EliminaTag
View Aggiunta tag	Attributi di un tag	AggiungiTag, ModificaTag
View Autenticazione	Nome utente e password	Autenticazione
View Registrazione	Nome utente, password e e-mail	Registrazione

La home page si apre sulla vista del ricettario, in alto a destra è disponibile il tasto per eseguire la ricerca. Un menù a scorrimento laterale permette di scegliere tra le varie funzionalità.

Analisi Ruoli e Responsabilità

Tabella dei ruoli

Ruolo	Responsabilità	Maschere	Riservatezza	Numerosità
Utente	Visualizzare e modificare i dati riguardanti gli ingredienti, i tag, le ricette, il frigo, lista della spesa e il calendario	Tutte le maschere lato client	La sensibilità delle informazioni è molto bassa	Uno per istanza dell'applicazione

Tabella Ruolo-Informazioni

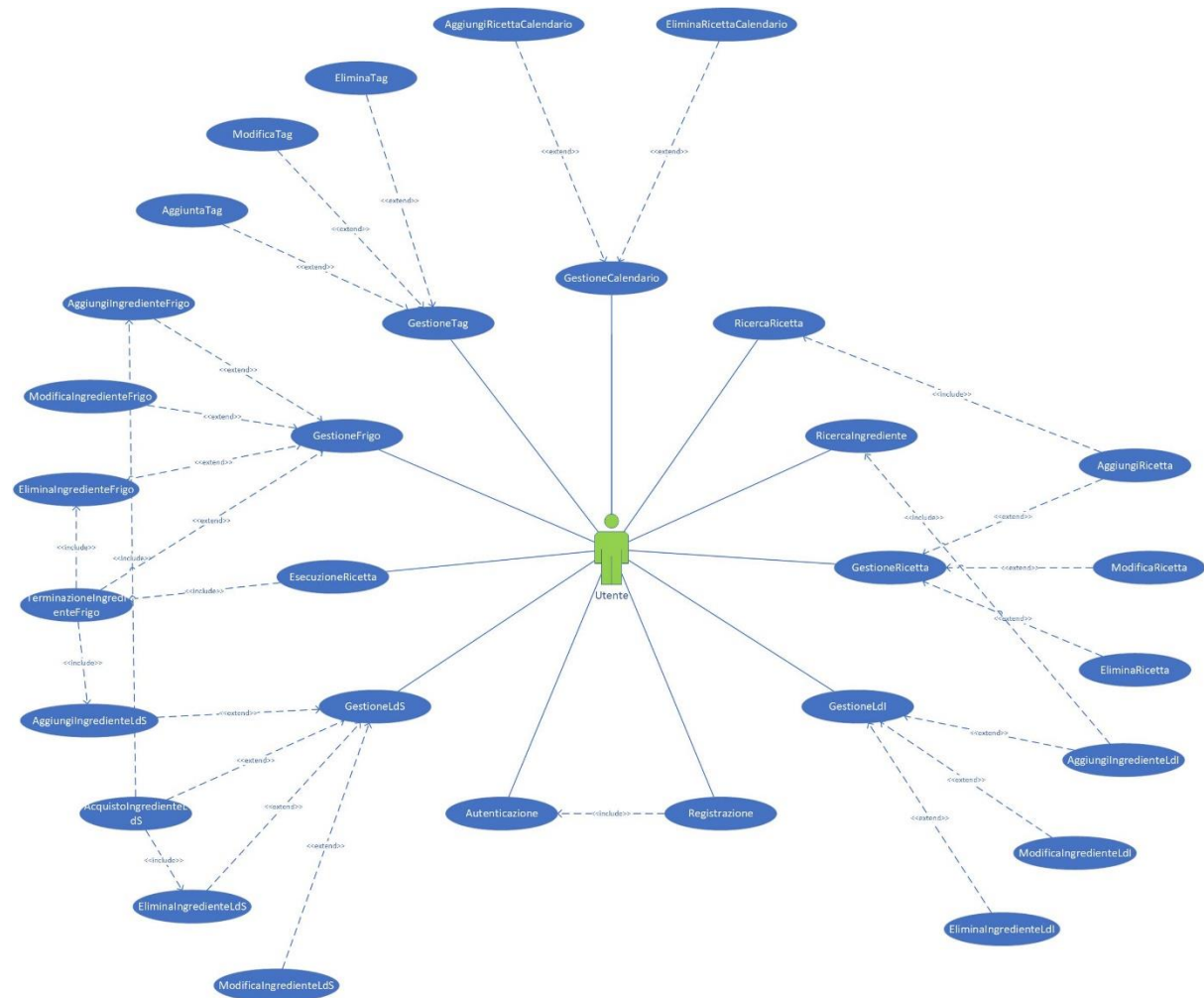
Informazione	Tipo di accesso
Ricetta	Lettura/Scrittura
Ingrediente	Lettura/Scrittura
Tag	Lettura/Scrittura
Frigo	Lettura/Scrittura
Lista della Spesa	Lettura/Scrittura
Calendario	Lettura/Scrittura

Scomposizione del problema

Tabella scomposizione funzionalità

Funzionalità	Scomposizione
GestioneRicetta	Aggiunta/Eliminazione/Modifica di una ricetta
GestioneFrigo	Aggiunta/Terminazione di un ingrediente dal frigo
GestioneLdS	Aggiunta/Eliminazione di un ingrediente dalla lista
GestioneLdl	Aggiunta/Eliminazione di un ingrediente dalla lista degli ingredienti di una ricetta
GestioneCalendario	Aggiunta/Eliminazione pasto dal calendario
GestioneTag	Aggiunta/Eliminazione di un tag da una ricetta

Analisi delle sottofunzionalità



AggiungiRicetta

Descrizione	Aggiunta ricetta
Attori	Utente
Relazioni	GestioneRicetta (<i>extends</i>)
Precondizioni	L'utente che parte con un ricettario di base, fornito dall'applicazione, può aggiungere una ricetta personale all'insieme delle ricette.
Postcondizioni	Viene aggiunta una ricetta personale nell'apposita sezione
Scenari principali	<ol style="list-style-type: none"> 1. L'utente aggiunge la ricetta con i relativi tag. 2. Il sistema registra il timestamp dato che potrà essere aggiornata in seguito
Scenari alternativi	Se esiste già una ricetta con lo stesso nome viene evidenziato il campo del nome per essere modificato. Gli altri campi rimangono salvati
Requisiti non funzionali	
Punti aperti	Cosa succede se inserisco una ricetta identica a una già esistente? E se cambia di poco? Deve essere l'utente a sapere se basta modificarla o se deve aggiungere una nuova?

ModificaRicetta

Descrizione	Modifica ricetta
Attori	Utente
Relazioni	GestioneRicetta (<i>extends</i>)
Precondizioni	Almeno una ricetta presente.
Postcondizioni	Viene aggiornata la ricetta nell'apposita sezione con un nuovo timestamp
Scenari principali	<ol style="list-style-type: none">1. L'utente può modificare una ricetta qualsiasi, aggiungendo e/o togliendo ingredienti, aggiungendo e/o togliendo o aggiungendo/eliminando passaggi.2. Il sistema registra il nuovo timestamp, che volendo potrà essere rimodificato
Scenari alternativi	Se esiste già una ricetta con lo stesso nome viene evidenziato il campo del nome per essere modificato. Gli altri campi rimangono salvati
Requisiti non funzionali	
Punti aperti	Dobbiamo tenere una "storia" delle modifiche? In questo caso la ricetta potrebbe essere anche completamente eliminata?

EliminaRicetta

Descrizione	Eliminazione ricetta
Attori	Utente
Relazioni	GestioneRicetta (<i>extends</i>)
Precondizioni	Almeno una ricetta presente.
Postcondizioni	Viene aggiornato l'insieme della ricetta, eliminando quella selezionata
Scenari principali	<ol style="list-style-type: none">1. L'utente, con un apposito tasto, può eliminare tutti i campi di una ricetta.2. inviato un messaggio di conferma della eliminazione
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

AggiungiIngredienteLdI

Descrizione	Aggiunta di un ingrediente alla lista degli ingredienti
Attori	Utente
Relazioni	GestioneLDI (<i>extends</i>)
Precondizioni	nessuna
Postcondizioni	Viene mostrato l'elenco degli ingredienti dell'utente
Scenari principali	<ol style="list-style-type: none"> 1. L'applicazione mostra all'utente i dettagli della ricetta e un pulsante per assegnarla a un pasto come nella visualizzazione di una ricetta 2. L'utente tocca il pulsante 3. L'applicazione mostra una vista per scegliere il giorno a cui assegnare la ricetta 4. L'utente sceglie il giorno 5. L'applicazione mostra l'elenco dei pasti del giorno con le eventuali ricette assegnate 6. L'utente tocca sul pasto al quale vuole assegnare una ricetta 7. L'applicazione assegna la ricetta al pasto selezionato 8. L'applicazione comunica all'utente che l'assegnamento è stato effettuato 9. L'applicazione mostra i dettagli della ricetta appena assegnata
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

AggiungiIngredienteLdS

Descrizione	Aggiunta di un ingrediente alla lista della spesa
Attori	Utente
Relazioni	GestioneLdS, Ricerca di un ingrediente (<i>include</i>)
Precondizioni	L'ingrediente deve essere presente nel database degli ingredienti
Postcondizioni	L'ingrediente viene aggiunto alla lista della spesa
Scenari principali	<ol style="list-style-type: none"> 1. L'applicazione mostra all'utente i dettagli della ricetta e un pulsante per assegnarla a un pasto come nella Visualizzazione di una ricetta 2. L'utente tocca il pulsante 3. L'applicazione mostra una vista per scegliere il giorno a cui assegnare la ricetta 4. L'utente sceglie il giorno 5. L'applicazione mostra l'elenco dei pasti del giorno con le eventuali ricette assegnate 6. L'utente tocca sul pasto al quale vuole assegnare una ricetta 7. L'applicazione assegna la ricetta al pasto selezionato 8. L'applicazione comunica all'utente che l'assegnamento è stato effettuato 9. L'applicazione mostra i dettagli della ricetta appena assegnata
Scenari alternativi	Scenario a: Ingrediente già presente nella lista della spesa <ol style="list-style-type: none"> 1. L'applicazione comunica all'utente che l'ingrediente è già presente nella lista della spesa 2. L'applicazione mostra una vista come quella iniziale
Requisiti non funzionali	Reattività della ricerca anche con molti ingredienti
Punti aperti	

ModificaIngredienteLdS

Descrizione	Modifica della quantità di un ingrediente nella lista della spesa
Attori	Utente
Relazioni	GestioneLDS (<i>extends</i>)
Precondizioni	L'ingrediente deve essere presente nella lista della spesa
Postcondizioni	La quantità dell'ingrediente da acquistare viene modificata
Scenari principali	<ol style="list-style-type: none"> 1. L'applicazione mostra l'elenco degli ingredienti nella lista della spesa 2. L'utente tocca sull'ingrediente del quale modificare la quantità da acquistare 3. L'applicazione mostra la quantità di quell'ingrediente attualmente registrata (o un campo vuoto se non è già definita) 4. L'utente tocca sulla quantità attualmente registrata 5. L'applicazione apre la modifica del campo di testo 6. L'utente inserisce la quantità da acquistare (o lascia vuoto il campo se non definita) e preme invio 7. L'applicazione salva la nuova quantità da acquistare 8. L'applicazione comunica all'utente che l'inserimento è avvenuto con successo 9. L'applicazione mostra l'elenco degli ingredienti nella lista della spesa
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

AcquistaIngredienteLdS

Descrizione	Acquisto di un ingrediente
Attori	Utente
Relazioni	EliminaIngredienteLdS (<i>include</i>), AggiungiIngredienteFrigo (<i>include</i>)
Precondizioni	L'utente ha almeno un ingrediente nella lista della spesa
Postcondizioni	Viene visualizzata la lista della spesa, l'ingrediente acquistato è depennato
Scenari principali	<ol style="list-style-type: none"> 1. L'utente spunta l'ingrediente da acquistare. 2. L'ingrediente è aggiunto al frigo.
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

TerminazioneIngredienteFrigo

Descrizione	Terminazione di un ingrediente
Attori	Utente
Relazioni	EsecuzioneRicetta, AggiungiIngredienteLdS (<i>include</i>), GestioneFrigo (<i>extends</i>) EliminaIngredienteFrigo (<i>include</i>)
Precondizioni	Nessuna
Postcondizioni	Gli ingredienti terminati sono aggiunti alla lista della spesa, e se erano nel frigo vengono rimossi.
Scenari principali	<ol style="list-style-type: none"> 1. L'applicazione mostra l'elenco degli ingredienti presenti nel frigo e un pulsante per rimuoverlo. 2. L'utente tocca sul pulsante 3. L'applicazione chiede all'utente se vuole rimuoverlo o aggiungerlo alla lista della spesa, contestualmente alla dose. 4. L'utente se vuole inserisce la dose, e invia la conferma. 5. L'applicazione rimuove l'ingrediente dal frigo e lo aggiunge alla lista della spesa. 6. L'applicazione mostra gli ingredienti presenti nel frigo.
Scenari alternativi	
Requisiti non funzionali	Comodità nell'interazione
Punti aperti	

AggiungiIngredienteFrigo

Descrizione	Aggiunta di un ingrediente al frigo
Attori	Utente
Relazioni	GestioneFrigo (<i>extends</i>)
Precondizioni	L'ingrediente deve essere presente nel databse degli ingredienti
Postcondizioni	L'ingrediente viene aggiunto al frigo
Scenari principali	<ol style="list-style-type: none"> 1. L'applicazione mostra l'elenco degli ingredienti presenti nel frigo e un pulsante per aggiungere un ingrediente 2. L'utente tocca sul pulsante 3. L'applicazione mostra una vista con un campo attraverso il quale cercare un ingrediente attraverso il suo nome 4. L'utente inserisce nel campo il nome dell'ingrediente 5. L'applicazione presenta un elenco di ingredienti risultato della ricerca 6. L'utente tocca l'ingrediente che vuole aggiungere al frigo tra quelli mostrati 7. L'applicazione inserisce nella lista della spesa l'ingrediente selezionato 8. L'applicazione comunica all'utente che l'inserimento è avvenuto con successo 9. L'applicazione mostra una vista come al punto 1
Scenari alternativi	Scenario a: Ingrediente già presente nel frigo <ol style="list-style-type: none"> 1. L'applicazione comunica all'utente che l'ingrediente è già presente nel frigo 2. L'applicazione mostra una vista come quella iniziale
Requisiti non funzionali	
Punti aperti	

AggiungiRicettaCalendario

Descrizione	Aggiunta di una ricetta al calendario
Attori	Utente
Relazioni	RicercaRicetta (<i>include</i>), GestioneCalendario (<i>extends</i>)
Precondizioni	La ricetta deve essere presente nel database delle ricette
Postcondizioni	La ricetta viene aggiunta al pasto selezionato
Scenari principali	<p>Da Calendario</p> <ol style="list-style-type: none"> 1. L'applicazione mostra il calendario della settimana/mese 2. L'utente tocca il giorno di suo interesse 3. L'applicazione mostra l'elenco dei pasti con le eventuali ricette assegnate ad ognuno 4. L'utente tocca sul pasto al quale vuole assegnare una ricetta 5. L'applicazione mostra la ricetta attualmente assegnata (se presente) e un pulsante per assegnare una ricetta 6. L'utente tocca sul pulsante 7. L'applicazione mostra una vista dalla quale cercare una ricetta 8. L'utente ricerca una ricetta come nel caso d'uso RicercaRicetta 9. L'applicazione mostra un elenco di ricette compatibili con la ricerca 10. L'utente tocca sulla ricetta da assegnare 11. L'applicazione assegna al giorno e al pasto selezionati la ricetta scelta 12. L'applicazione comunica all'utente che l'assegnamento è stato effettuato 13. L'applicazione mostra una vista come al punto 1
Scenari alternativi	<p>Dalla visualizzazione della ricetta:</p> <ol style="list-style-type: none"> 1. L'applicazione mostra all'utente i dettagli della ricetta e un pulsante per assegnarla a un pasto come nella Visualizzazione di una ricetta 2. L'utente tocca il pulsante 3. L'applicazione mostra una vista per scegliere il giorno a cui assegnare la ricetta 4. L'utente sceglie il giorno 5. L'applicazione mostra l'elenco dei pasti del giorno come le eventuali ricette assegnate 6. L'utente tocca sul pasto al quale vuole assegnare una ricetta 7. L'applicazione assegna la ricetta al pasto selezionato 8. L'applicazione comunica all'utente che l'assegnamento è stato effettuato 9. L'applicazione mostra i dettagli della ricetta appena assegnata
Requisiti non funzionali	Leggibilità dell'interfaccia. Facilità di navigazione nell'interfaccia
Punti aperti	

EliminaIngredienteLdS

Descrizione	Rimozione di un ingrediente dalla lista della spesa
Attori	Utente
Relazioni	GestioneLdS (<i>Extends</i>)
Precondizioni	L'ingrediente deve essere presente nella lista della spesa
Postcondizioni	L'ingrediente viene rimosso dalla lista della spesa
Scenari principali	<ol style="list-style-type: none">1. L'applicazione mostra l'elenco degli ingredienti nella lista della spesa2. L'utente tocca sull'ingrediente che vuole rimuovere3. L'applicazione mostra i dettagli dell'ingrediente e un tasto attraverso il quale rimuoverlo4. L'utente tocca sul pulsante5. L'applicazione rimuove l'ingrediente dalla lista della spesa6. L'applicazione comunica all'utente che la rimozione è avvenuto con successo7. L'applicazione mostra l'elenco degli ingredienti nella lista della spesa
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

EliminaRicettaCalendario

Descrizione	Rimozione di una ricetta assegnata a un pasto dal calendario
Attori	Utente
Relazioni	GestioneCalendario (<i>extends</i>)
Precondizioni	La ricetta deve essere assegnata a un determinato pasto in un determinato giorno
Postcondizioni	La ricetta viene tolta dal pasto selezionato
Scenari principali	<ol style="list-style-type: none">1. L'applicazione mostra il calendario della settimana/mese2. L'utente tocca il giorno di suo interesse3. L'applicazione mostra l'elenco dei pasti con le eventuali ricette assegnate ad ognuno4. L'utente tocca sul pasto dal quale vuole rimuovere la ricetta5. L'applicazione mostra la ricetta attualmente assegnata e un pulsante per rimuoverla6. L'utente tocca sul pulsante7. L'applicazione rimuove la ricetta dal pasto8. L'applicazione comunica all'utente che la rimozione è stata effettuata9. L'applicazione mostra una vista come al punto 1
Scenari alternativi	
Requisiti non funzionali	Leggibilità dell'interfaccia. Facilità di navigazione nell'interfaccia
Punti aperti	

AggiungiTag

Descrizione	Aggiunta di un tag
Attori	Utente
Relazioni	GestioneTag (<i>extends</i>)
Precondizioni	
Postcondizioni	Il tag viene aggiunto a quelli già esistenti
Scenari principali	<ol style="list-style-type: none">1. L'applicazione mostra l'elenco dei tag.2. L'utente tocca sul pulsante per aggiungere un nuovo tag3. L'utente inserisce i dati del tag mediante l'apposita interfaccia.4. L'utente invia la conferma.5. Se non esiste già lo stesso tag, l'applicazione lo aggiunge alla lista, altrimenti lo segnala all'utente e resta nel menù di modifica.6. L'applicazione mostra la vista del punto uno.
Scenari alternativi	
Requisiti non funzionali	Reattività della ricerca anche con molti tag
Punti aperti	

ModificaTag

Descrizione	Modifica di un tag
Attori	Utente
Relazioni	GestioneTag (<i>extends</i>)
Precondizioni	Almeno un tag presente.
Postcondizioni	Viene aggiornato il tag
Scenari principali	<ol style="list-style-type: none">1. L'utente visualizza l'elenco dei tag2. L'utente tocca il tag da modificare ed effettua le modifiche tramite l'interfaccia3. L'utente salva la modifica e viene riportato all'elenco dei tag
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

ModificaIngredienteLdl

Descrizione	Modifica di un ingrediente
Attori	Utente
Relazioni	GestioneIngredienti (<i>extends</i>)
Precondizioni	Almeno un ingrediente presente.
Postcondizioni	Viene aggiornato il tag
Scenari principali	<ol style="list-style-type: none">1. L'utente visualizza l'elenco degli ingredienti2. L'utente tocca l'ingrediente da modificare ed effettua le modifiche tramite l'interfaccia3. L'utente salva la modifica e viene riportato all'elenco degli ingredienti
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

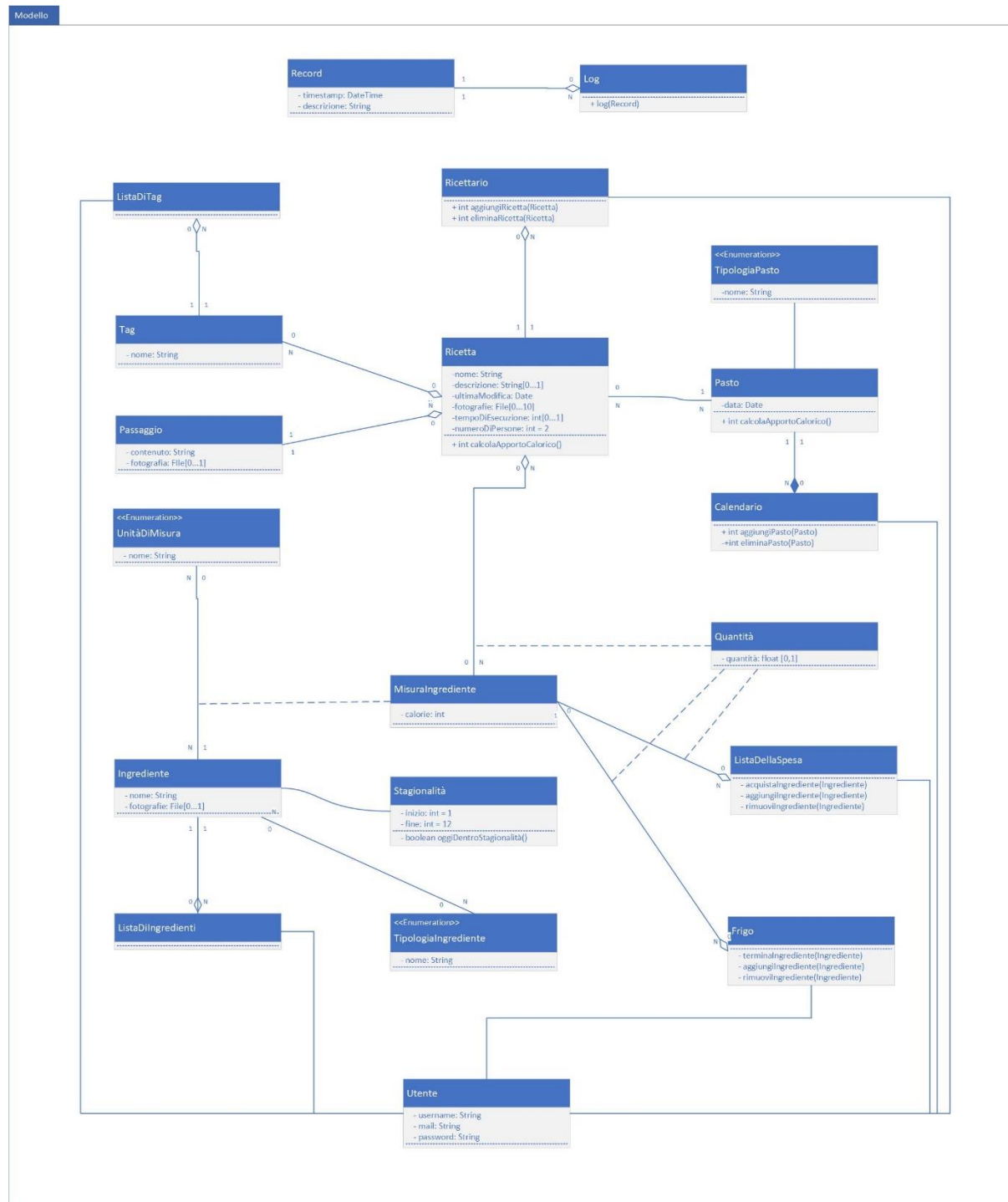
EliminaTag

Descrizione	Eliminazione di un tag
Attori	Utente
Relazioni	GestioneTag (<i>extends</i>)
Precondizioni	Almeno un tag presente.
Postcondizioni	Il tag scelto viene eliminato
Scenari principali	<ol style="list-style-type: none">1. L'utente visualizza l'elenco dei tag2. L'utente elimina il tag attraverso l'apposito pulsante3. Il tag viene eliminato e l'utente riceve un messaggio di conferma
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	Cosa fare con le ricette a cui è assegnato quel tag? Bloccare l'eliminazione se è in uso mantenendo un contatore delle ricette che usano ogni tag?

EliminaIngredienteLdl

Descrizione	Rimozione di un ingrediente dalla lista degli ingredienti
Attori	Utente
Relazioni	GestioneIngredienti (<i>extends</i>)
Precondizioni	L'ingrediente deve essere presente nella lista degli ingredienti
Postcondizioni	L'ingrediente viene rimosso dalla lista degli ingredienti
Scenari principali	<ol style="list-style-type: none">1. L'applicazione mostra l'elenco degli ingredienti nella lista degli ingredienti2. L'utente tocca sull'ingrediente che vuole rimuovere3. L'applicazione mostra i dettagli dell'ingrediente e un tasto attraverso il quale rimuoverlo4. L'utente tocca sul pulsante5. L'applicazione rimuove l'ingrediente dalla lista degli ingredienti6. L'applicazione comunica all'utente che la rimozione è avvenuto con successo7. L'applicazione mostra l'elenco degli ingredienti nella lista degli ingredienti
Scenari alternativi	
Requisiti non funzionali	
Punti aperti	

Modello del dominio



Tutti gli elementi del ricettario affluiscono a un solo utente: se anche due utenti hanno la stessa identica ricetta, per il sistema sono due ricette diverse.

MisuraIngrediente indica che tra un ingrediente e un'unità di misura ci può essere una sola associazione, che viene usata nelle ricette, nel frigo e nella lista della spesa.

Quantità indica che all'interno dell'associazione Ricetta – MisuraIngrediente è possibile associare una quantità opzionale. Discorso analogo vale per Frigo e ListaDellaSpesa.

Architettura logica

Struttura

Diagramma dei package



Diagramma delle classi: EsecuzioneRicetta, GestioneFrigo e GestioneLds

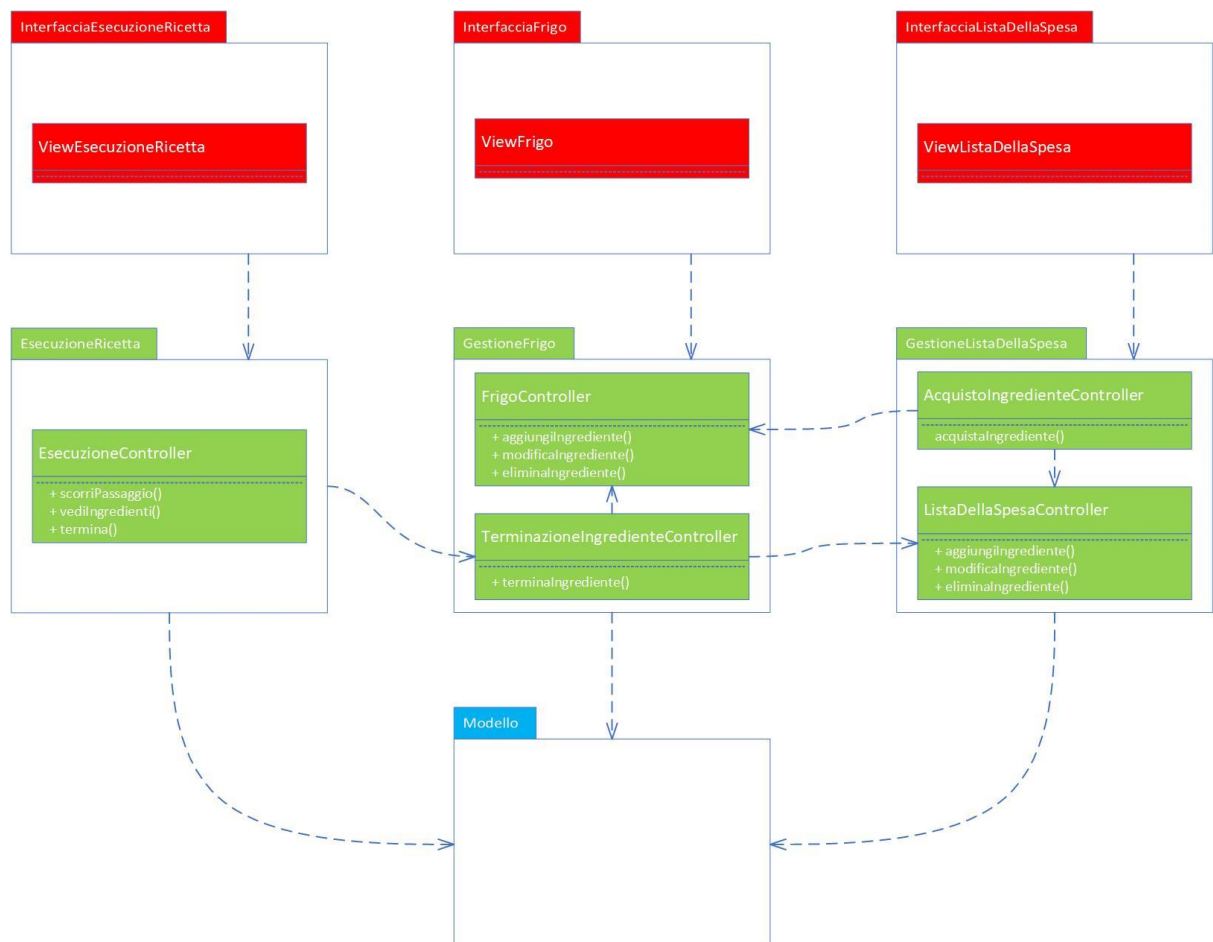


Diagramma delle classi: GestioneCalendario

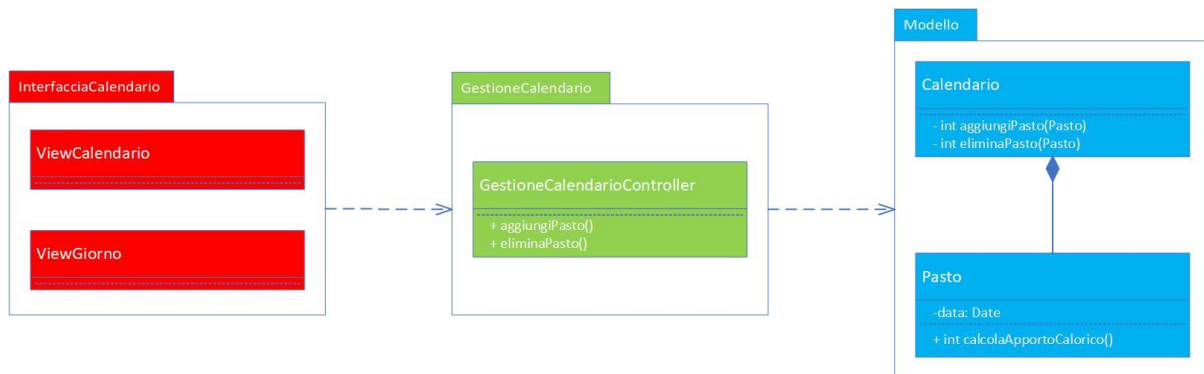


Diagramma delle classi: GestioneLdl

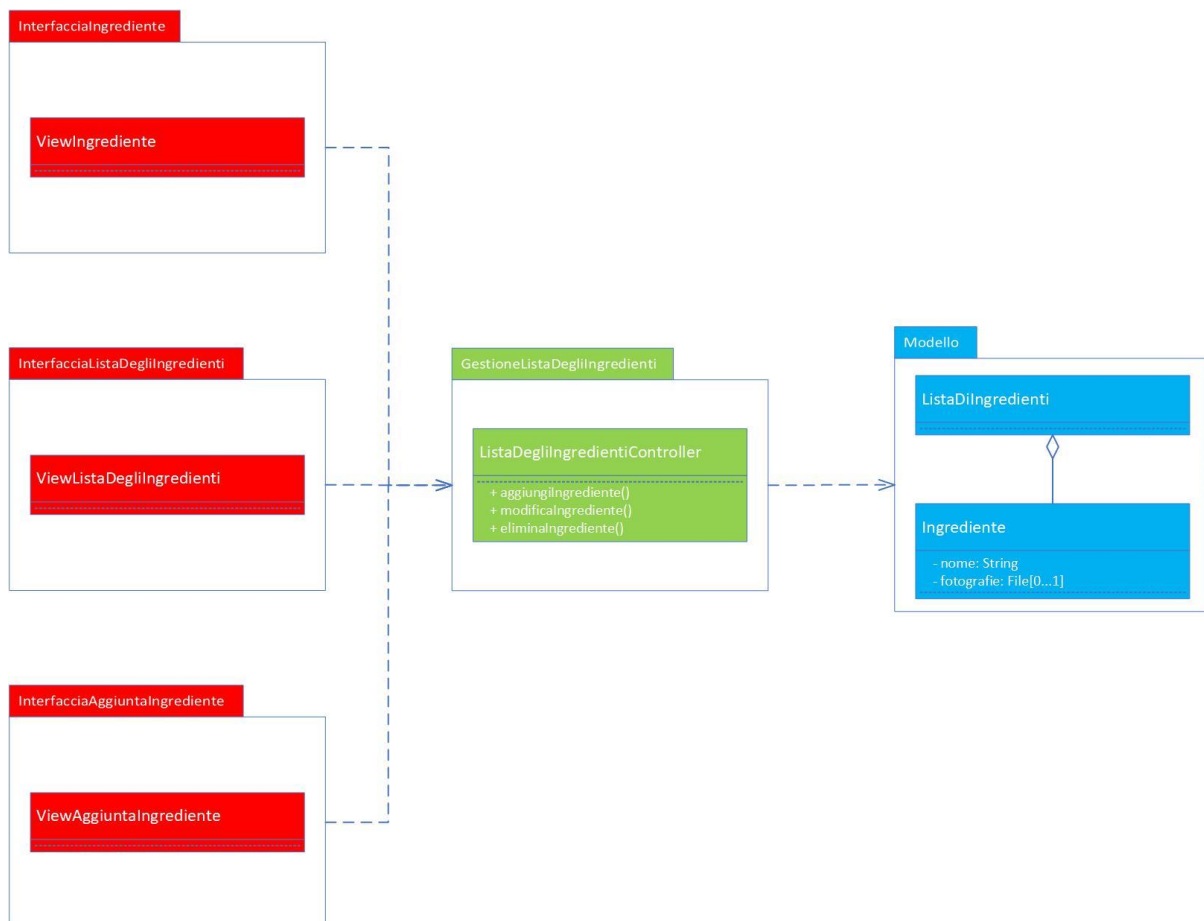


Diagramma delle classi: GestioneRicettario

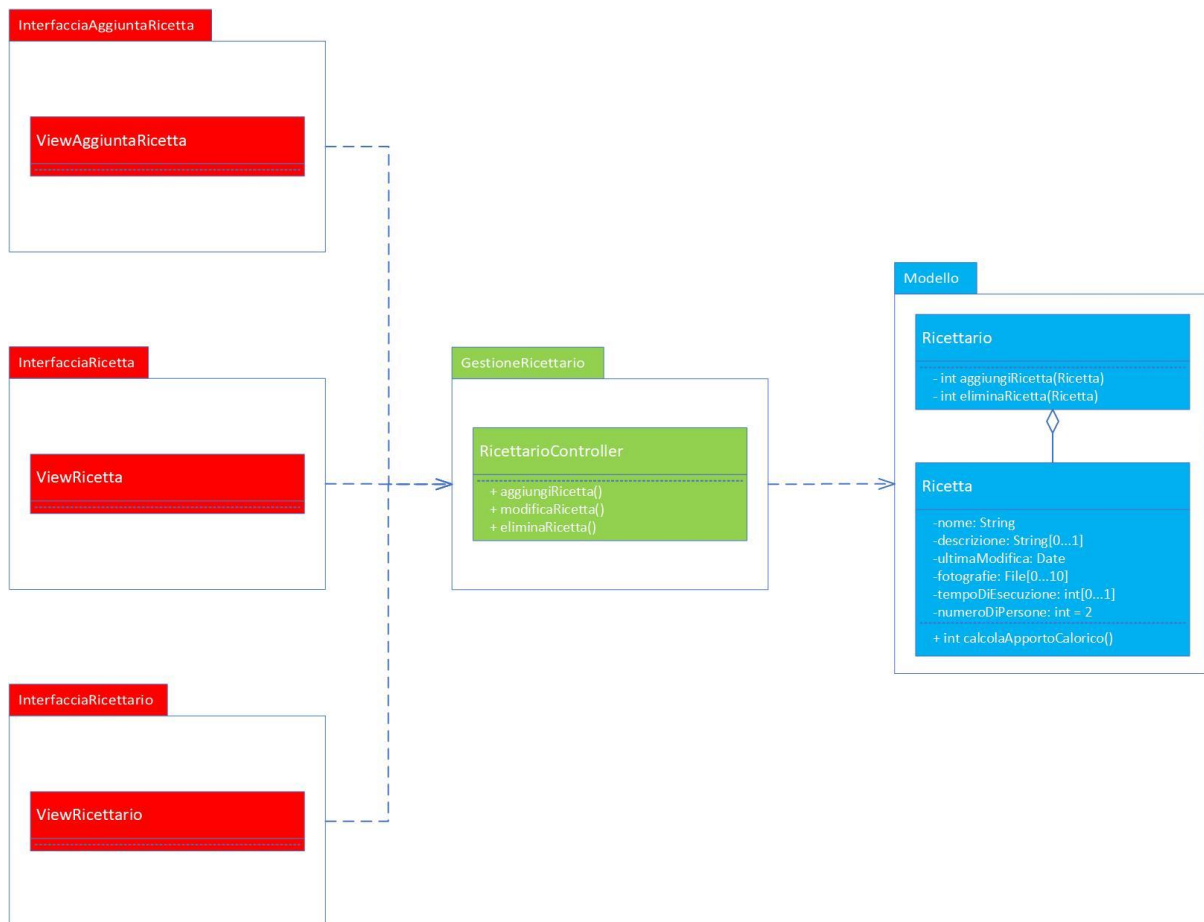


Diagramma delle classi: GestioneTag

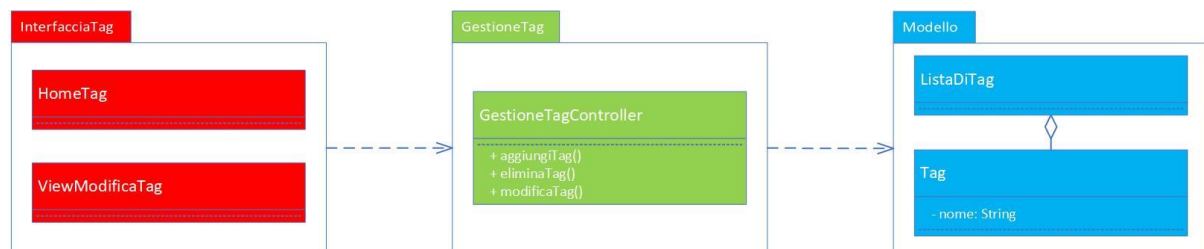


Diagramma delle classi: RicercaIngrediente



Diagramma delle classi: RicercaRicetta

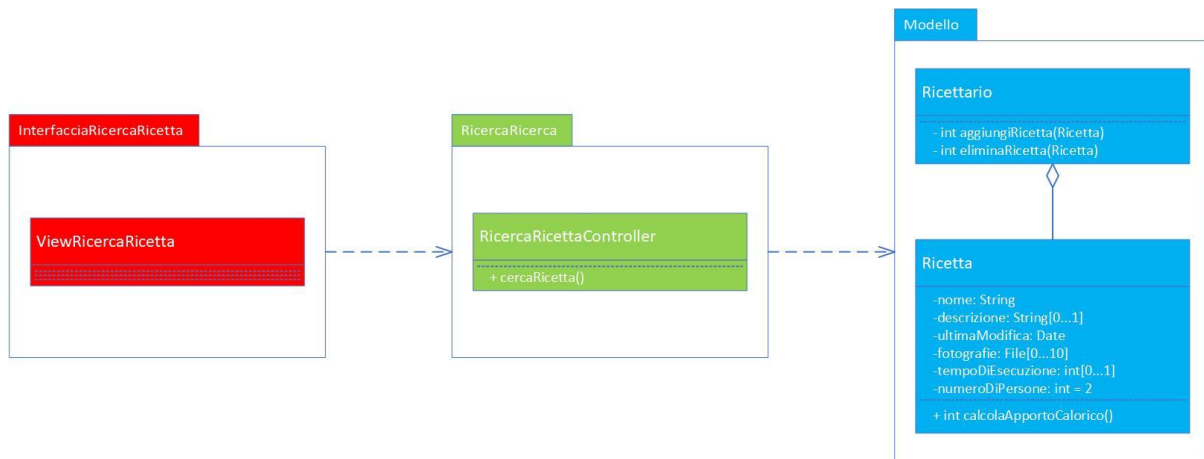
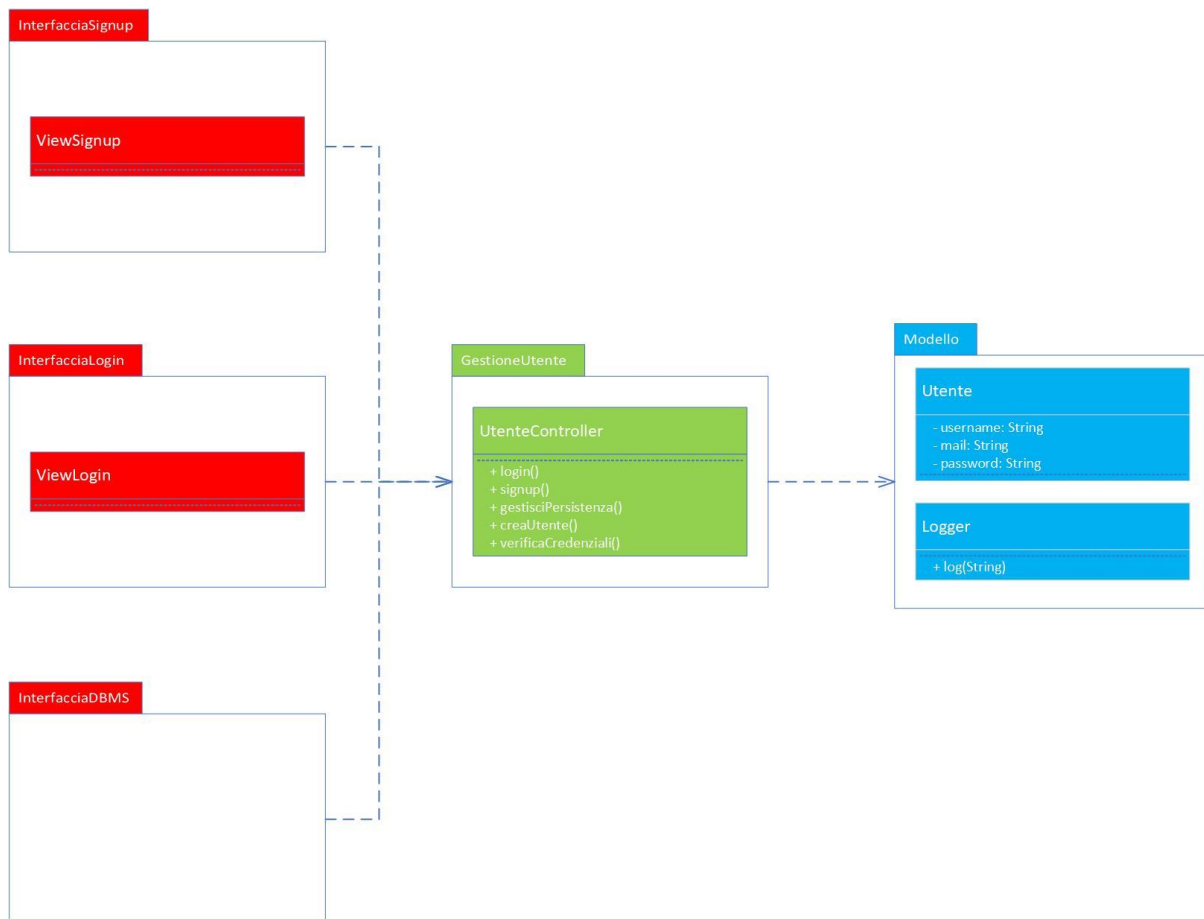


Diagramma delle classi: GestioneUtente



Interazione

Diagramma di sequenza: gestione ingredienti

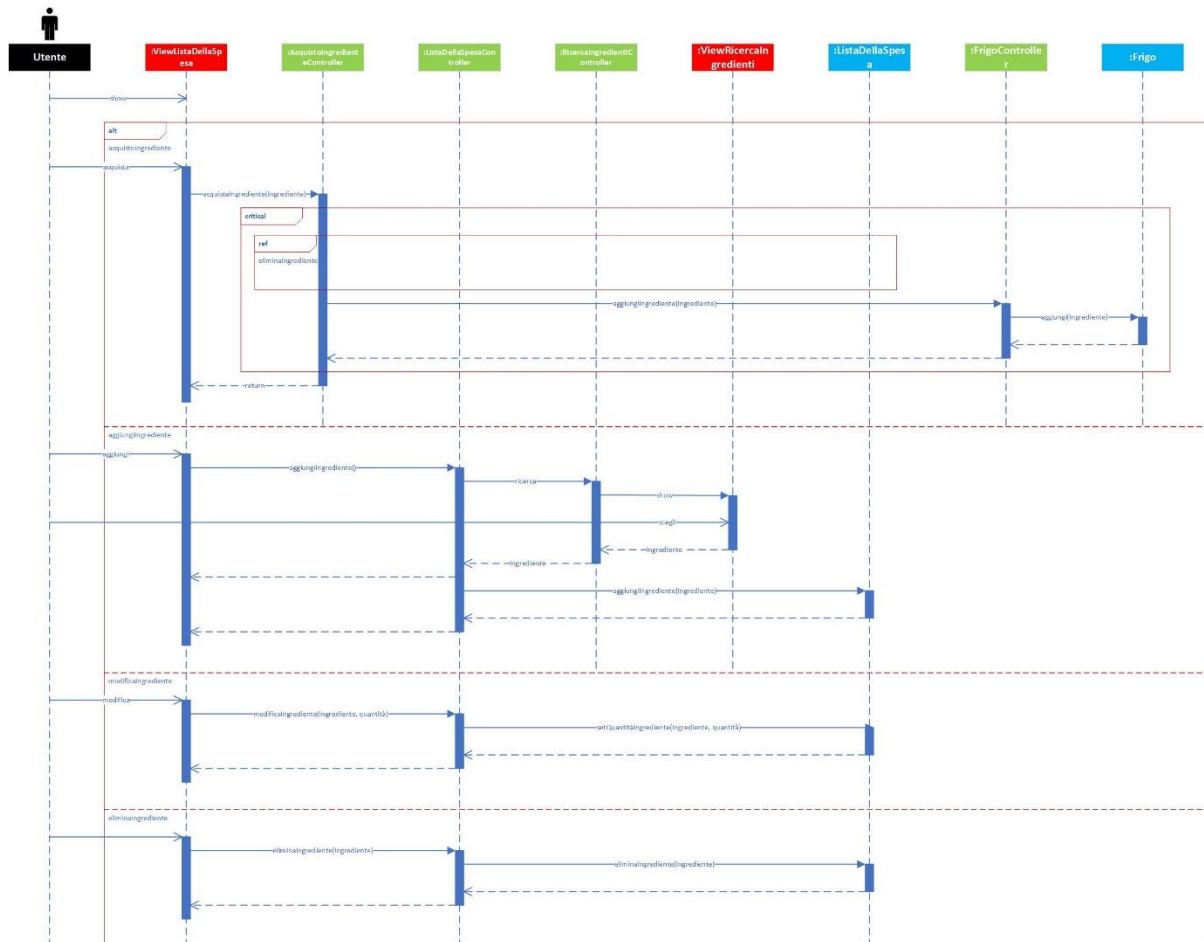


Diagramma di sequenza: registrazione eseguita con successo

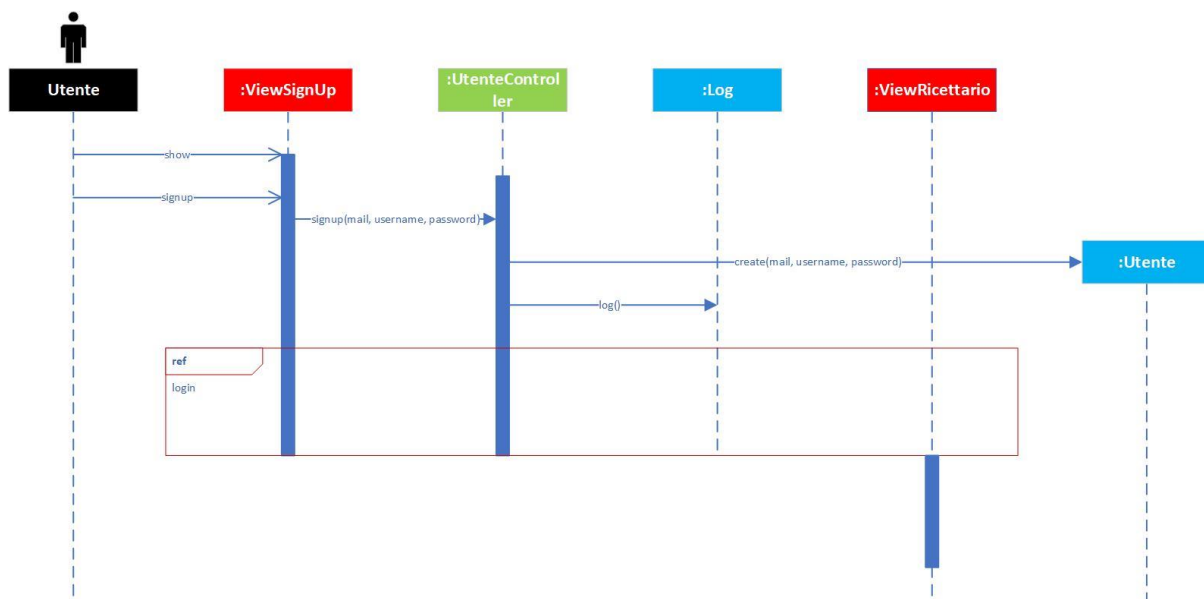
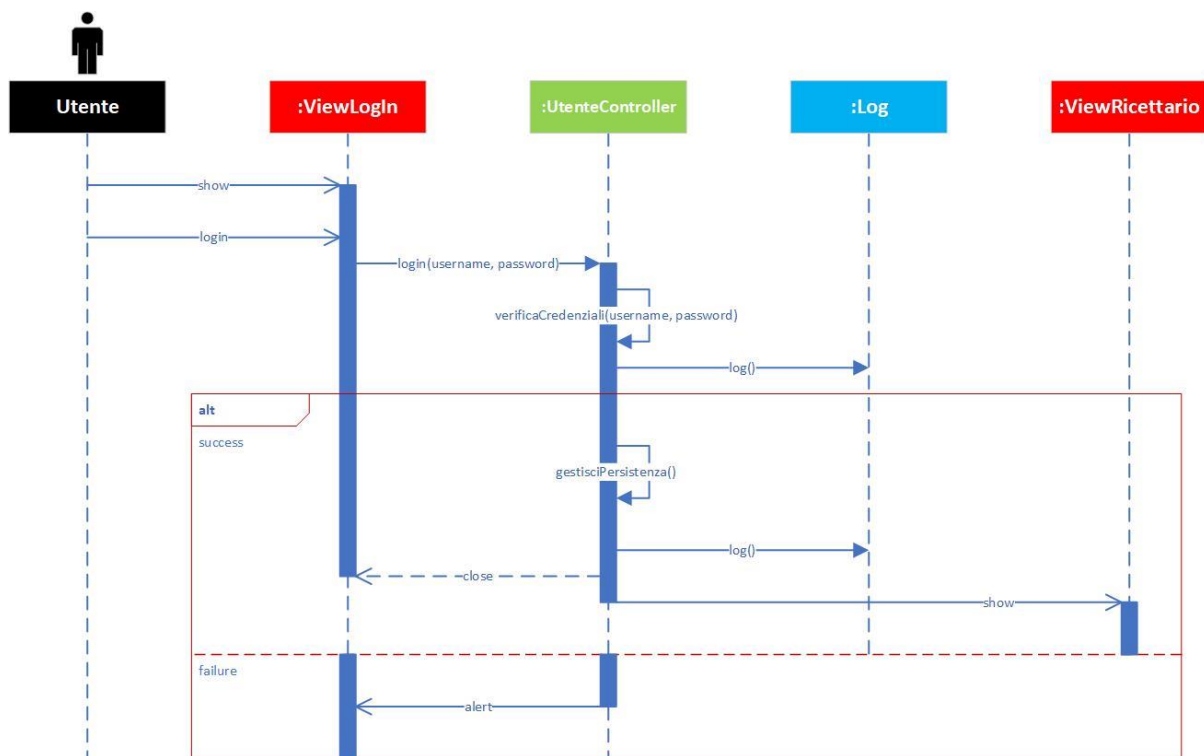
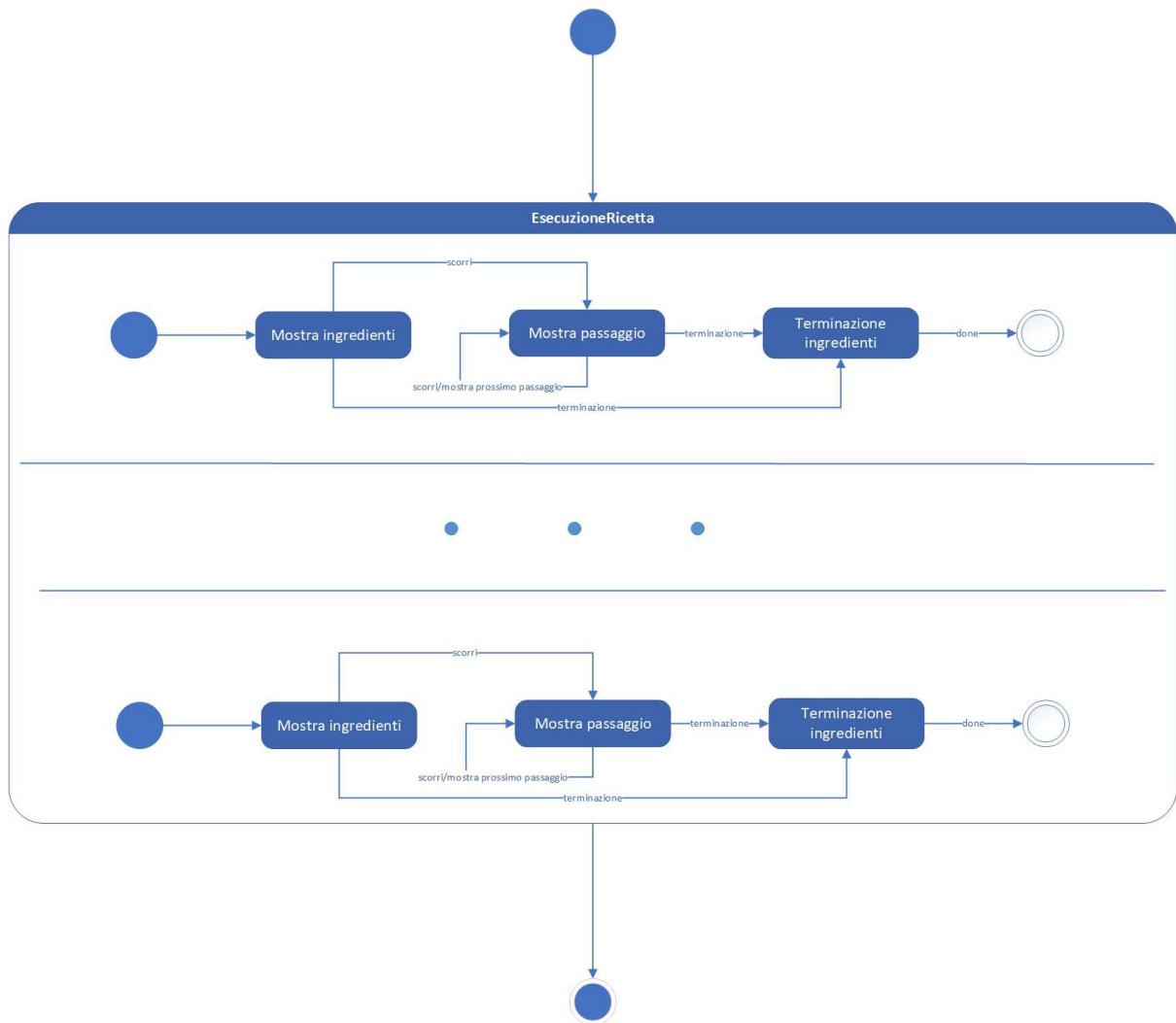


Diagramma di sequenza: accesso eseguito con successo



Comportamento

Diagramma di stato: EsecuzioneRicetta



Piano di lavoro

Package	Progetto	Sviluppo
Modello	Bartoli, Medici, Olivo	Bartoli, Medici, Olivo
Log	Bartoli, Medici, Olivo	Bartoli, Medici
GestioneTag	Bartoli, Medici, Olivo	Bartoli, Medici
RicercaIngrediente	Bartoli, Medici, Olivo	Bartoli, Medici
GestioneCalendario	Bartoli, Medici, Olivo	Bartoli, Medici
GestioneListaDegliIngredienti	Bartoli, Medici, Olivo	Bartoli, Medici
GestioneRicettario	Bartoli, Medici, Olivo	Bartoli, Medici
RicercaRicetta	Bartoli, Medici, Olivo	Bartoli, Medici
GestioneFrigo	Bartoli, Medici, Olivo	Bartoli, Medici
GestioneListaDellaSpesa	Bartoli, Medici, Olivo	Bartoli, Medici
EsecuzioneRicetta	Bartoli, Medici, Olivo	Bartoli, Medici
GestioneUtente	Bartoli, Medici, Olivo	Bartoli, Medici
InterfacciaTag	Bartoli, Medici, Olivo	Olivo
InterfacciaRicercaIngrediente	Bartoli, Medici, Olivo	Olivo
InterfacciaCalendario	Bartoli, Medici, Olivo	Olivo
InterfacciaListaDegliIngredienti	Bartoli, Medici, Olivo	Olivo
InterfacciaIngrediente	Bartoli, Medici, Olivo	Olivo
InterfacciaAggiuntaIngrediente	Bartoli, Medici, Olivo	Olivo
InterfacciaRicettario	Bartoli, Medici, Olivo	Olivo
InterfacciaRicetta	Bartoli, Medici, Olivo	Olivo
InterfacciaAggiuntaRicetta	Bartoli, Medici, Olivo	Olivo
InterfacciaRicercaRicerca	Bartoli, Medici, Olivo	Olivo
InterfacciaFrigo	Bartoli, Medici, Olivo	Olivo
InterfacciaListaDellaSpesa	Bartoli, Medici, Olivo	Olivo
InterfacciaEsecuzioneRicetta	Bartoli, Medici, Olivo	Olivo
InterfacciaLogin	Bartoli, Medici, Olivo	Olivo
InterfacciaSignup	Bartoli, Medici, Olivo	Olivo
InterfacciaDBMS	Bartoli, Medici, Olivo	Olivo

Sviluppi futuri

Per il futuro si prevede di implementare nuovi metodi per aggiungere una ricetta, da fotocamera, da siti etc.

Si prevede inoltre di inserire una funzionalità di condivisione di ricette tra utenti registrati, tramite la quale si potrà inviare ad un altro utente una propria ricetta o ingrediente. Questo giustifica una progettazione così dettagliata della persistenza lato server.

Piano di collaudo

```
public class TestIngrediente {

    private Ingrediente ingrediente;

    @Before
    public void setUp() {
        ingrediente = new Ingrediente();
    }

    @Test
    public void TestCostruttore() {
        Assert.assertNull(ingrediente.getNome());
        Assert.assertNull(ingrediente.getfotografia());
    }

    @Test
    public void TestGetterProprietà() {
        ingrediente = new Ingrediente("Pasta", new File("fotoPasta.jpg"));
        Assert.assertEquals("Pasta", ingrediente.getNome());
        Assert.assertEquals(new File("fotoPasta.jpg"), ingrediente.getfotografia());
    }

    @Test
    public void TestSetterProprietà() {
        ingrediente.setNome("Prosciutto");
        Assert.assertEquals("Prosciutto", ingrediente.getNome());
        ingrediente.setfotografia(new File("fotoProsciutto.jpg"));
        Assert.assertEquals(new File("fotoProsciutto.jpg"), ingrediente.getfotografia());
    }
}
```

```

public class TestRicetta {

    private Ricetta ricetta;

    @Before
    public void setUp() {
        ricetta = new Ricetta();
    }

    @Test
    public void TestCostruttore() {
        Assert.assertNull(ricetta.getNome());
        Assert.assertNull(ricetta.getDescrizione());
        Assert.assertNull(ricetta.getUltimaModifica());
        Assert.assertNull(ricetta.getFotografie());
        Assert.assertEquals(-1, ricetta.getTempoDiEsecuzione());
        Assert.assertEquals(2, ricetta.getNumeroDiPersone());
    }

    @Test
    public void TestGetterProprietà() {
        LocalDate edit = LocalDate.now();
        File[] foto = new File[10];
        foto[0] = new File("fotoPastaPesto1.jpg");
        ricetta = new Ricetta("Pasta al pesto", "Una buona pasta al pesto", edit, foto, 20, 2);
        Assert.assertEquals("Pasta al pesto", ricetta.getNome());
        Assert.assertEquals("Una buona pasta al pesto", ricetta.getDescrizione());
        Assert.assertEquals(edit, ricetta.getUltimaModifica());
        Assert.assertArrayEquals(foto, ricetta.getFotografie());
        Assert.assertEquals(20, ricetta.getTempoDiEsecuzione());
        Assert.assertEquals(2, ricetta.getNumeroDiPersone());
    }

    @Test
    public void TestSetterProprietà() {
        ricetta.setNome("Uova al tegamino");
        Assert.assertEquals("Uova al tegamino", ricetta.getNome());
        ricetta.setDescrizione("Un delizioso seconod piatto");
        Assert.assertEquals("Un delizioso seconod piatto", ricetta.getDescrizione());
        LocalDate edit = LocalDate.now().minusDays(1);
        ricetta.setUltimaModifica(edit);
        Assert.assertEquals(edit, ricetta.getUltimaModifica());
        File[] foto = new File[10];
        foto[0] = new File("fotoUovaTegamino1.jpg");
        ricetta.setFotografie(foto);
        Assert.assertArrayEquals(foto, ricetta.getFotografie());
        ricetta.setTempoDiEsecuzione(5);
        Assert.assertEquals(5, ricetta.getTempoDiEsecuzione());
        ricetta.setNumeroDiPersone(1);
        Assert.assertEquals(1, ricetta.getNumeroDiPersone());
    }
}

```

Progettazione

Progettazione architetturale

Requisiti non funzionali

Dall'analisi del problema risulta chiara la presenza di alcuni requisiti non funzionali, tipici delle applicazioni per dispositivi mobili:

- Usabilità
- Prestazioni
- Integrità dei dati
- Manutenibilità

Nel nostro caso, se da un lato usabilità e prestazioni richiedono entrambe caratteristiche affini, quali componenti “grandi” che riducano comunicazioni quindi tempo di caricamento, d'altro canto la manutenibilità del sistema e la sua necessità di essere facilmente estendibile richiedono una riduzione in dimensione dei componenti. Sarà dunque necessario trovare un compromesso che possa garantire da un lato prestazioni rapide, che non compromettano l'interazione dell'utente, e facilità di estensioni. Soprattutto in fase iniziale si è deciso di focalizzarsi sulla manutenibilità, data l'alta probabilità che il feedback degli utenti in fase di rilascio conduca alla correzione di bug ed errori in fase di progettazione. In una fase secondaria, quando l'utenza si sarà stabilizzata e l'applicazione sarà il più fruibile possibile, in caso di prestazioni migliorabili si potrà rifattorizzare il codice prediligendo la velocità di risposta. Va inoltre sottolineato che in un mercato eterogeneo e in rapido sviluppo come quello dei dispositivi mobili, le prestazioni possono facilmente dipendere più che dall'applicazione dal software su cui è installata, fattore che potrebbe in parte vanificare uno sforzo eccessivo nel migliorare le prestazioni.

Scelta dell'architettura

Considerando gli sviluppi futuri, si è scelto di utilizzare un modello client/server a tre livelli, con la persistenza divisa in modo replicato tra client e server in ottica di disponibilità. Uno dei concetti fondamentali è di avere un client che possa lavorare sempre, indipendentemente dalla disponibilità del server. Il server, che verrà progettato in seguito, si dovrà carico di gestire dispositivi multipli associati a uno stesso account, garantendo l'integrità dei dati anche nei casi critici che possono incorrere. Analogamente anche la persistenza remota verrà progettata in futuro.

L1 – Client

Il client svolgerà la quasi totalità delle operazioni. Si potrà in futuro connettere al server mediante una connessione sicura così da garantire l'accesso da più dispositivi.

L2 – Server

Il server sarà unico e si occuperà di garantire la “concorrenza” tra più dispositivi, oltre che di mantenere una copia dei dati in remoto.

L3a – Persistenza locale

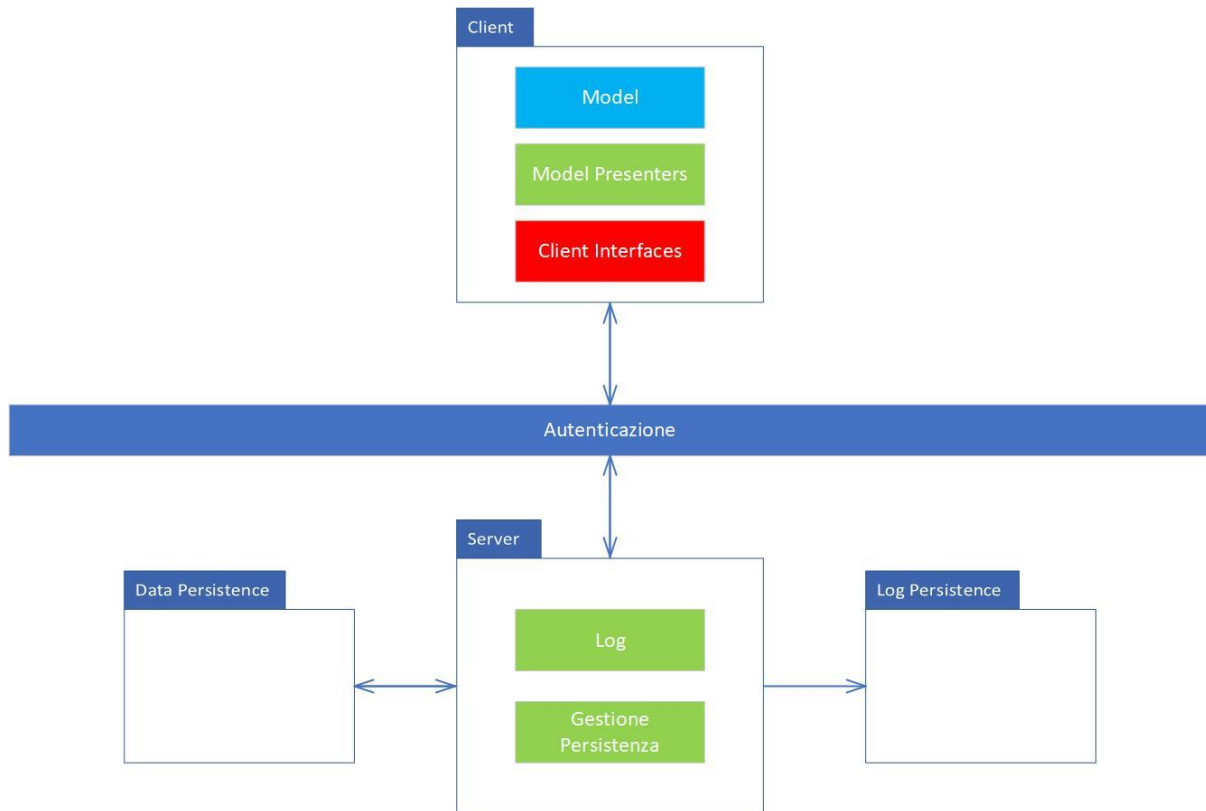
La persistenza locale sfrutterà la memoria fisica del dispositivo. Sarà in futuro replicabile sulla persistenza remota così da garantire l'accesso da più dispositivi.

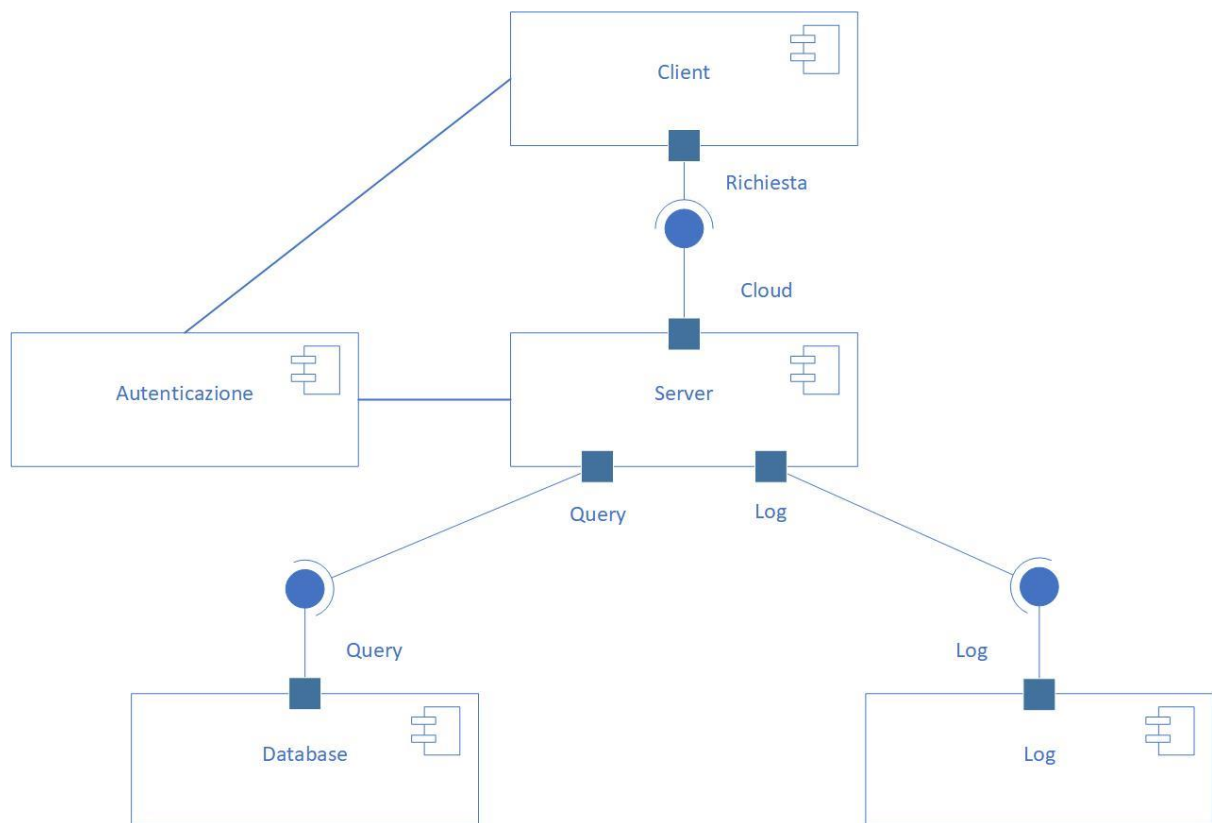
L3b – Persistenza remota

La persistenza remota sfrutterà un server dedicato su cui sarà installato un DBMS in grado di gestire i dati e mantenere la versione più adeguata tra quelle presenti sui vari dispositivi, e un server dedicato al log. In questo modo si separano i componenti legati alla gestione applicativa da quelli relativi alla

sicurezza. Le operazioni sul DBMS avverranno tramite i metodi CRUD, secondo la metodologia della “forza bruta”

Per garantire i requisiti di sicurezza desiderati la comunicazione tra Client e Server avverrà utilizzando il protocollo TLS (*Transport Layer Security*), un protocollo crittografico che permette comunicazioni sicure end-to-end su comunicazioni di tipo *stream* utilizzando lo stack di protocolli TCP/IP, utilizzando principalmente il meccanismo della cifratura.

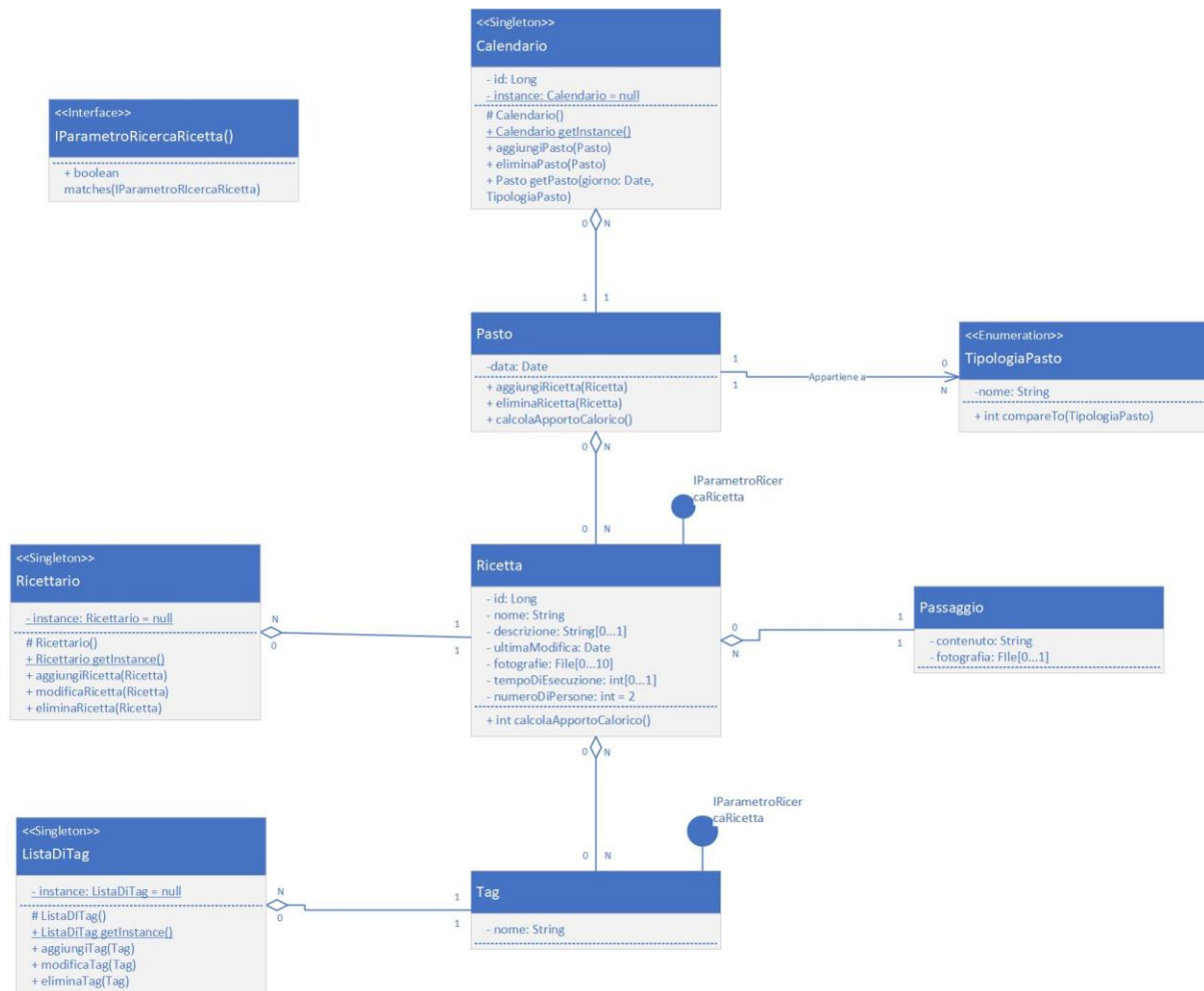




Scelte tecnologiche

Abbiamo deciso di progettare un'applicazione Android per il client, data la necessità di usare un linguaggio proprietario per i sistemi iOS. Il linguaggio utilizzato sarà quello Java, data la conoscenza comune a tutto il gruppo. Si era valutato anche Kotlin ma, data la scarsa manualità e le scadenze ristrette, si è prediletto il linguaggio maggiormente noto a tutto il gruppo.

Diagramma di dettaglio: dominio del client – Ricetta

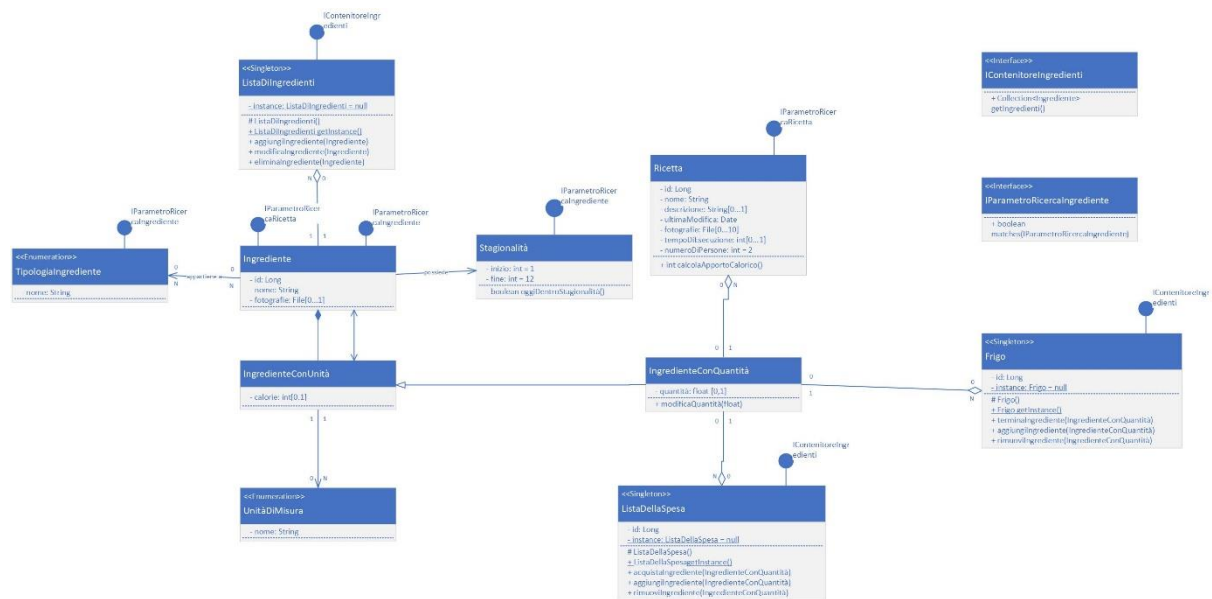


Rispetto al modello in fase di analisi si può notare come scompaia ogni informazione relativa all'utente, infatti il client è completamente agnostico dell'esistenza di altri client, e dunque di altri ricettari o di altri calendari. È aggiunta l'interfaccia marker **IParametroRicercaRicetta**, che verrà utilizzata per estendere la ricerca di una ricetta in futuro. Viene inoltre aggiunto un id a **Ricetta** e **Calendario**, id che verrà utilizzato per garantire la consistenza con il database a fronte di modifiche. Questo id verrà creato insieme all'istanza e non sarà modificabile.

Tutti gli elementi che sono singleton all'interno del client possiedono i metodi per realizzare il pattern.

Non serve che da **Tag** si possa navigare verso **Ricetta**, infatti è sufficiente andare da **Ricetta** a **Tag** in fase di ricerca.

Diagramma di dettaglio: dominio del client - Ingrediente



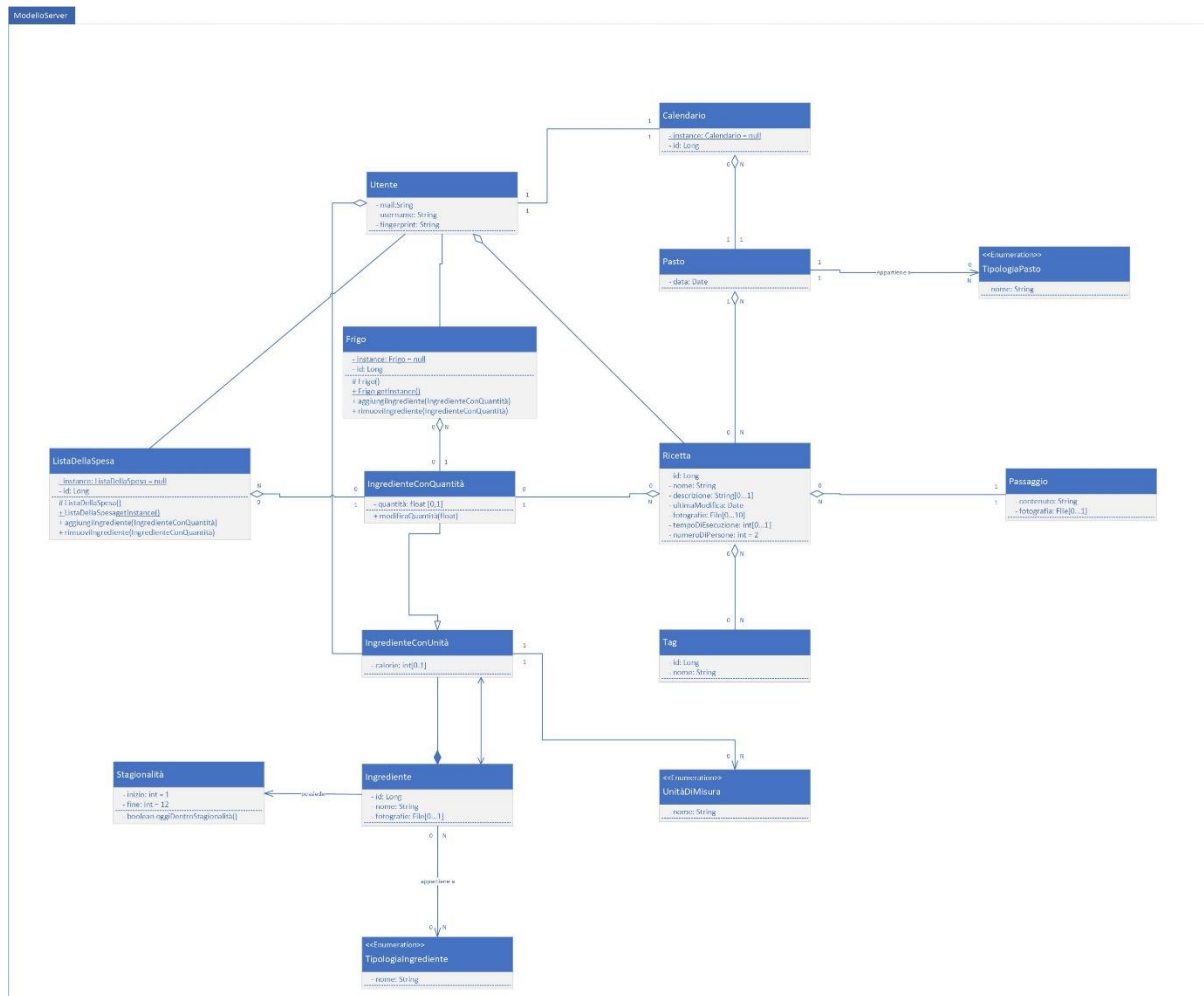
Anche qui sono stati aggiunti gli identificativi, per cui valgono considerazioni analoghe a quelle del caso precedente. Rispetto alla fase di analisi, la classe MisuraIngrediente diventa una classe di associazione e viene rinominata in IngredienteConUnità. Ingrediente diventa una composizione di IngredienteConUnità, e alla creazione di un Ingrediente, se l'utente non seleziona nessuna unità di misura, ne viene utilizzata una di default. L'associazione è bidirezionale per poter risalire da una Ricetta all'ingrediente.

Analogamente Quantità diventa IngredienteConUnità, che eredita da IngredienteConQuantità, a cui aggiunge opzionalmente la quantità. Ricetta, Frigo e ListaDellaSpesa usano questa classe.

Anche qui sono presenti le interfacce marker IContentoreDiIngrediente e IParametroRicercaIngrediente, utilizzate in fase di ricerca.

Un Pasto viene creato dal Calendario, ma si aggiunge a esso solo quando si aggiunge una ricetta al pasto. Il Pasto deve quindi mantenere un riferimento al calendario per potersi aggiungere.

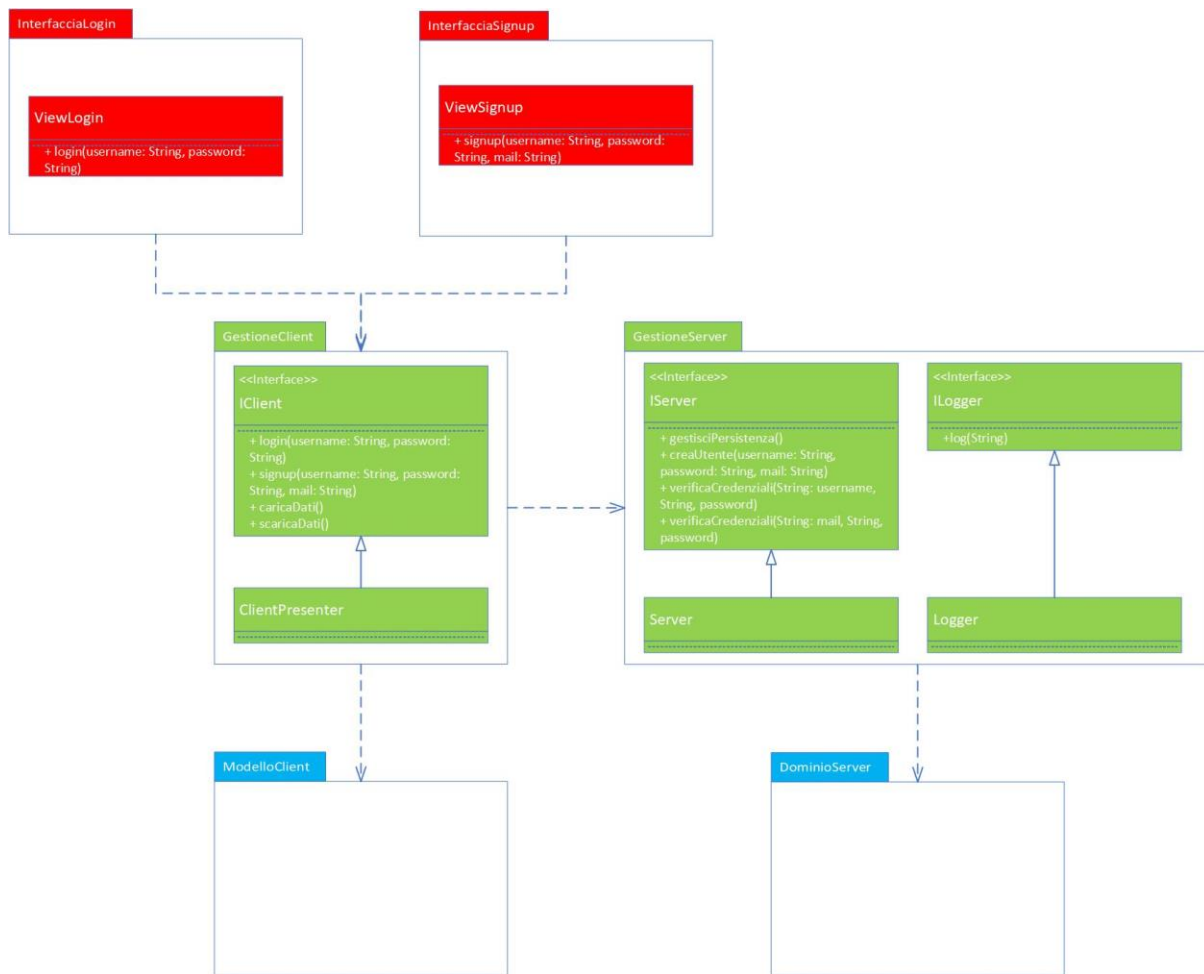
Diagramma di dettaglio: dominio del server



Rispetto al client le differenze sono la presenza dell'utente, la rimozione di metodi legati al dominio, qui non necessari, e delle classi contenitore come ListaDiTag o Ricettario, identificate dalla relazione Utente – Tag o Utente – Ricetta.

Sebbene non esplicitamente segnalati, tutte le classi del server contengono i metodi per implementare la persistenza mediante Forza Bruta.

Diagramma di dettaglio: Autenticazione, Registrazione e Gestione della persistenza



Sono aggiunte le interfacce `IClient`, `IServer` e `ILogger`, così da rispettare il Dependency Inversion Principle.

Sono state rimosse le interfacce Log e DBMS. La prima perché i file vengono acceduti direttamente, la seconda perché abbiamo scelto di accedere al database mediante Forza Bruta, di conseguenza il server comunica con il database direttamente attraverso il modello.

Diagramma di dettaglio: GestioneCalendario

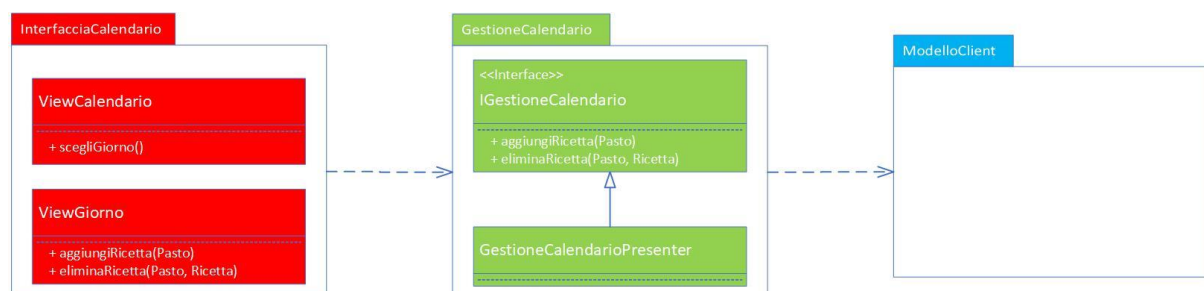


Diagramma di dettaglio: GestioneLdS, GestioneFrigo ed EsecuzioneRicetta



I package **AcquistoIngredienteController** e **TerminazioneIngredienteController** sono stati inglobati all'interno dei package principali, in quanto rappresentavano una sottofunzionalità troppo semplice da giustificare un package specifico.

Per risolvere la dipendenza ciclica tra **GestioneFrigo** e **GestioneLdS** in fase di analisi, abbiamo collocato l'interfaccia **IListaDellaSpesa** all'interno del package di **GestioneFrigo**, rispettando così il **Dependency Inversion Principle**. Il metodo dell'interfaccia **IEsecuzione** `calcolaRicetteEseguibili()` sfrutta le informazioni di sistema sulle dimensioni dello schermo per adattarsi e restare leggibile sia con schermi particolarmente ampi, come quello di un tablet, sia con quello più ridotto di uno smartphone.

Frigo e **ListaDellaSpesa** si dovranno comportare come insiemi di ingredienti, se viene aggiunto un ingrediente già presente la quantità dovrà essere sommata.

Il presenter dell'esecuzione deve tenere conto di quali ricette sono in esecuzione e del passaggio in corso, in modo trasparente alla view.

Diagramma di dettaglio: GestioneLdl e RicercaIngrediente

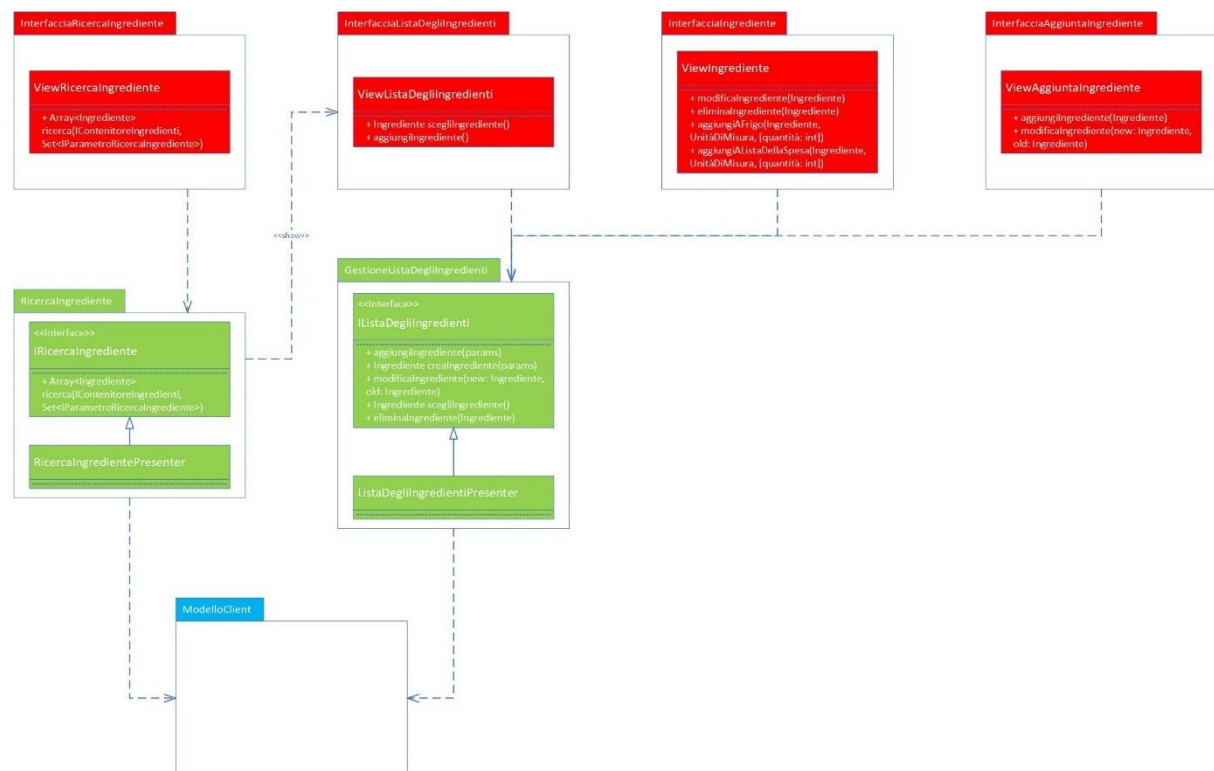
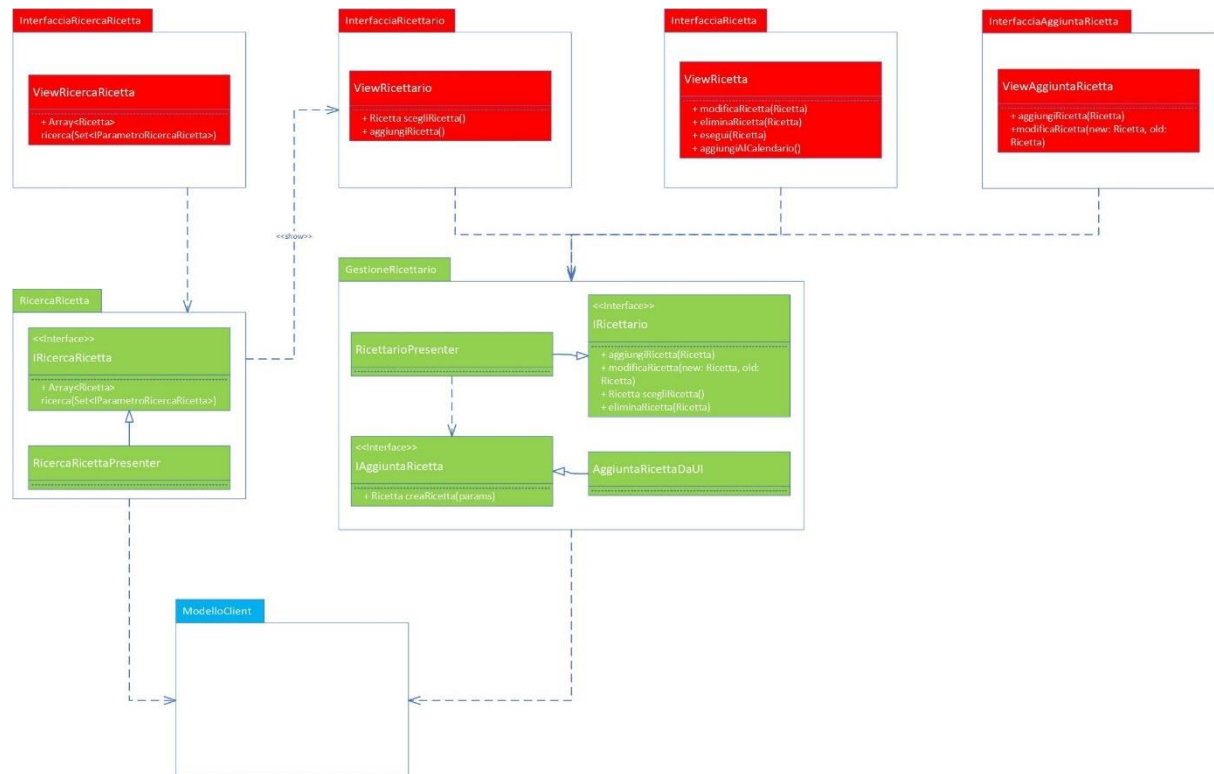
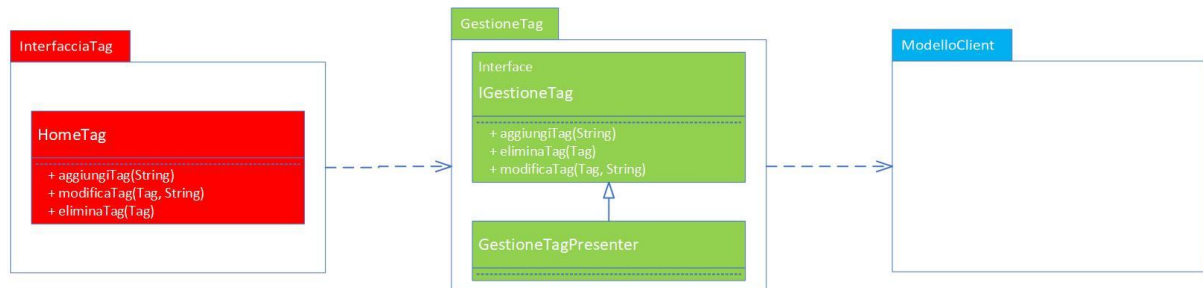


Diagramma di dettaglio: GestioneRicettario e RicercaRicetta



L'interfaccia `IAggiuntaRicetta` è stata fatta per soddisfare il requisito R6UF. In questo modo sarà sufficiente aggiungere una nuova implementazione dell'interfaccia per avere un nuovo modo di aggiungere una ricetta, sfruttando diversi metodi.

Diagramma di dettaglio: GestioneTag



La view `ModificaTag` è stata eliminata, l'operazione sarà fattibile in linea e non richiederà un'interfaccia apposita.

Interazione

Diagramma di sequenza: acquisto ingrediente

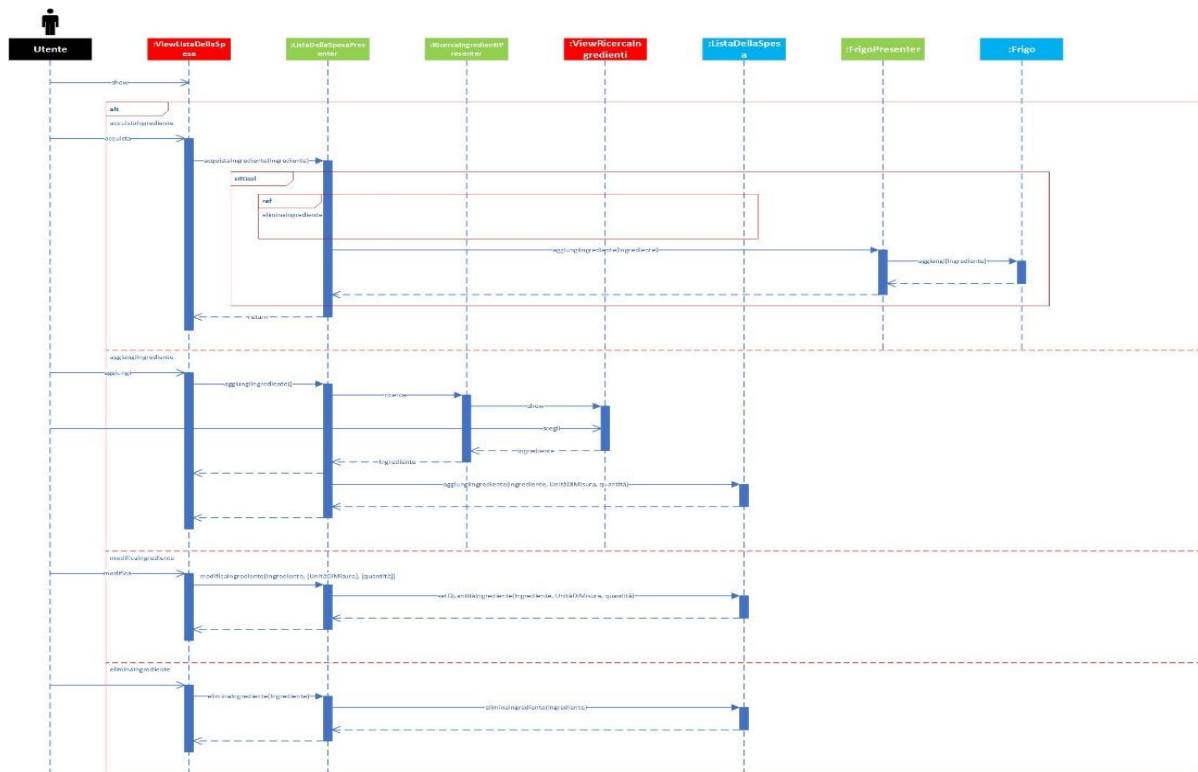


Diagramma di sequenza: login

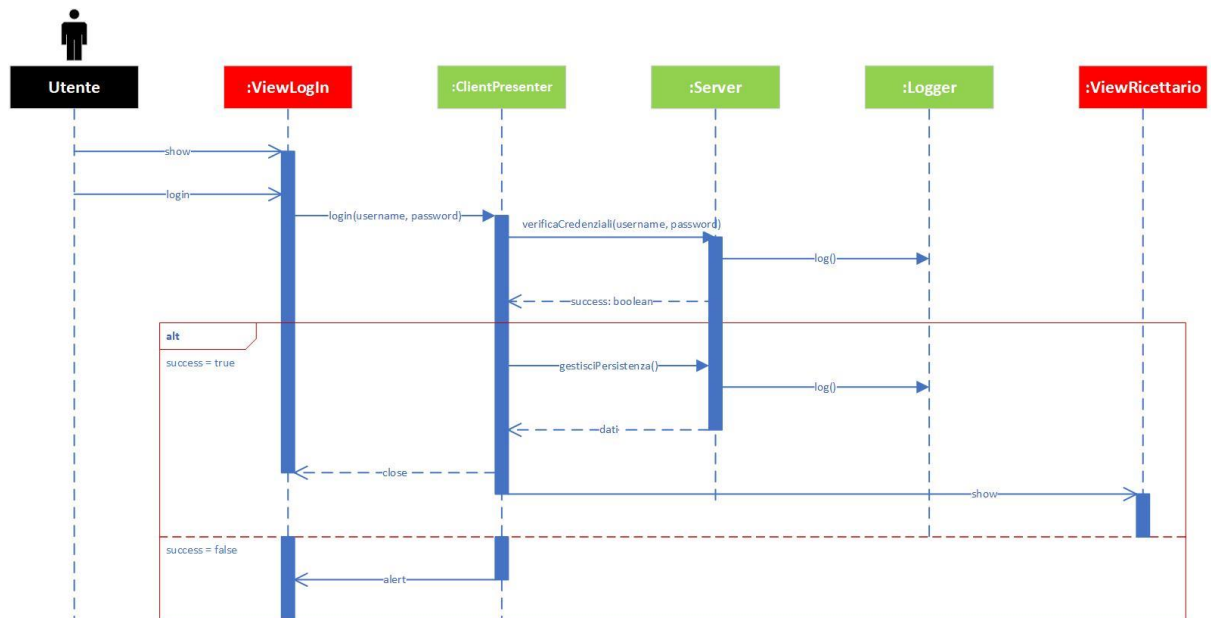
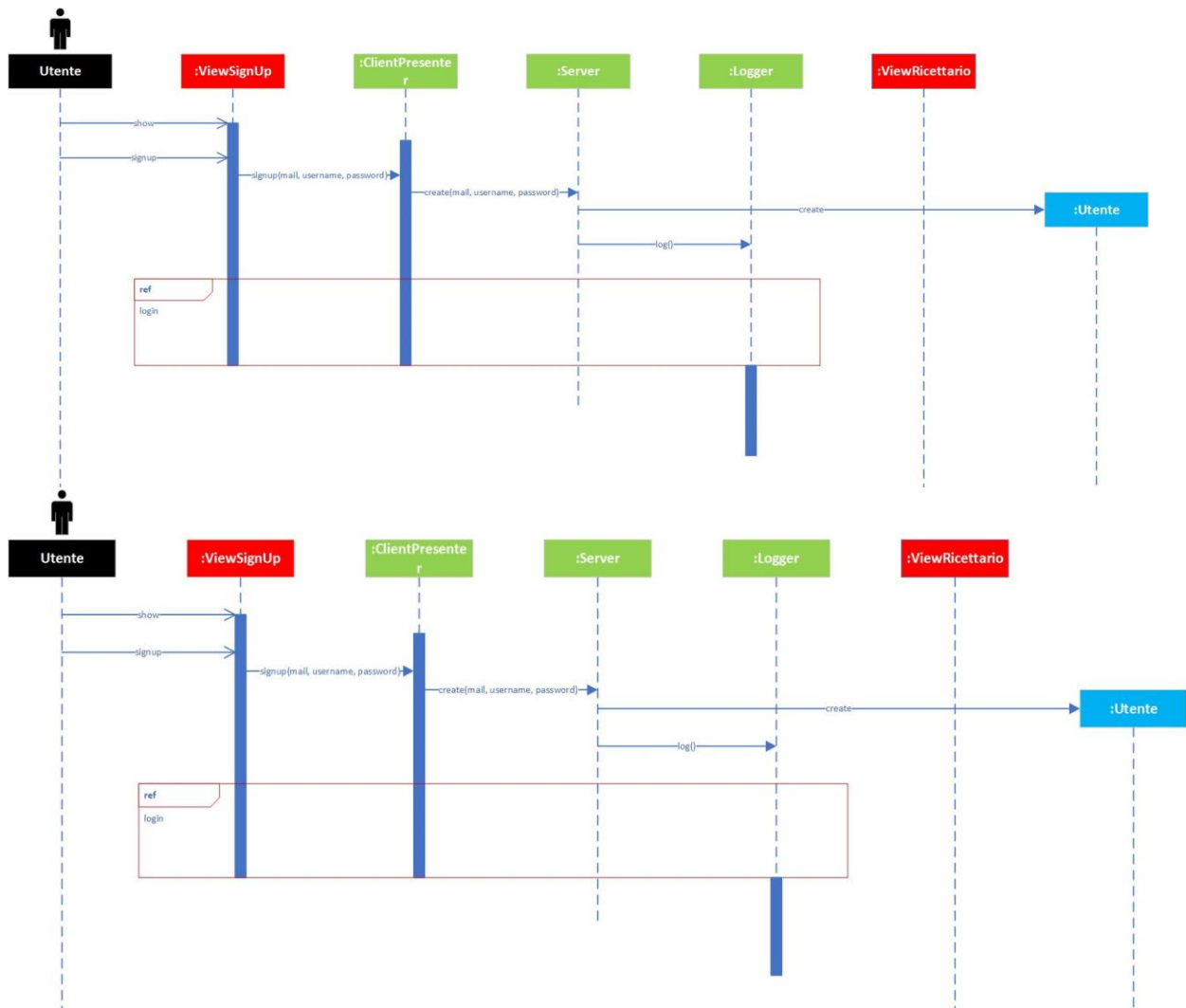


Diagramma di sequenza: signup



Comportamento

Riteniamo non siano necessari ulteriori diagrammi oltre a quello presente in fase di analisi del problema.

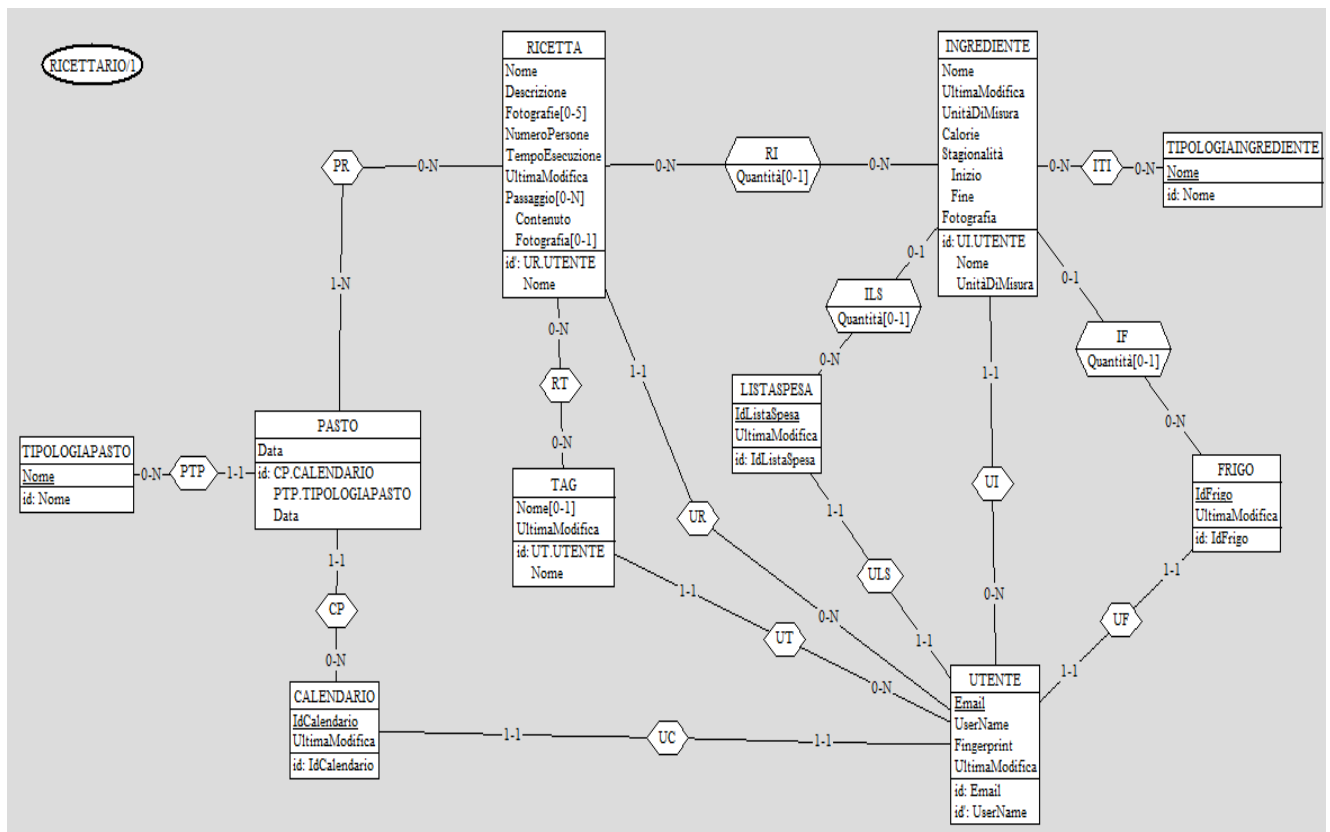
Progettazione persistenza

Utilizziamo un database relazionale per mantenere i dati omogenei.

Sono rappresentati nel diagramma sottostante tutte le entità e le relazioni della persistenza.

Abbiamo scelto di utilizzare un identificativo univoco per le entità per garantire consistenza dei dati, soprattutto nella fase di modifica dei nomi delle ricette, degli ingredienti o dei tag.

Diagramma ER:



Il log viene mantenuto su file, accessibile non dall'applicazione ma dagli amministratori del sistema.

Progettazione del collaudo

```
public class TestCalendario {

    private Calendario calendario;
    private Pasto pasto;

    @Before
    public void setUp() {
        calendario = Calendario.getInstance();
        pasto = calendario.getPasto(LocalDate.now(), TipologiaPasto.COLAZIONE);
        LocalDate edit = LocalDate.now();
        File[] foto = new File[10];
        foto[0] = new File("fotoPastaPesto1.jpg");
        IngredienteConQuantità[] ingredienti = new IngredienteConQuantità[2];
        ingredienti[0] = new IngredienteConQuantità(20, ListaDegliIngredienti.getIngrediente("Pasta"), UnitàDiMisura.GRAMMI, 200);
        ingredienti[1] = new IngredienteConQuantità(35, ListaDegliIngredienti.getIngrediente("Pesto"), UnitàDiMisura.GRAMMI, 50);
        pasto.aggiungiRicetta(new Ricetta("Pasta al pesto", "Una deliziosa pasta al pesto", edit, foto, 20, 2, ingredienti));
        calendario.aggiungiPasto(pasto);
    }

    @Test
    public void TestCostruttore() {
        Assert.assertEquals(Calendario.getInstance(), calendario);
        Pasto pastoTest = calendario.getPasto(LocalDate.now(), TipologiaPasto.COLAZIONE);
        LocalDate edit = LocalDate.now();
        File[] foto = new File[10];
        foto[0] = new File("fotoPastaPesto1.jpg");
        IngredienteConQuantità[] ingredienti = new IngredienteConQuantità[2];
        ingredienti[0] = new IngredienteConQuantità(20, ListaDegliIngredienti.getIngrediente("Pasta"), UnitàDiMisura.GRAMMI, 200);
        ingredienti[1] = new IngredienteConQuantità(35, ListaDegliIngredienti.getIngrediente("Pesto"), UnitàDiMisura.GRAMMI, 50);
        pastoTest.aggiungiRicetta(new Ricetta("Pasta al pesto", "Una deliziosa pasta al pesto", edit, foto, 20, 2, ingredienti));
        calendario.aggiungiPasto(pastoTest);
        Assert.assertEquals(pasto, calendario.getPasto(LocalDate.now(), TipologiaPasto.COLAZIONE));
    }

    @Test
    public void TestOperazioni() {
        Pasto pastoTest = calendario.getPasto(LocalDate.now(), TipologiaPasto.COLAZIONE);
        LocalDate edit = LocalDate.now();
        File[] foto = new File[10];
        foto[0] = new File("fotoPastaPesto1.jpg");
        IngredienteConQuantità[] ingredienti = new IngredienteConQuantità[2];
        ingredienti[0] = new IngredienteConQuantità(20, ListaDegliIngredienti.getIngrediente("Pasta"), UnitàDiMisura.GRAMMI, 200);
        ingredienti[1] = new IngredienteConQuantità(35, ListaDegliIngredienti.getIngrediente("Pesto"), UnitàDiMisura.GRAMMI, 50);
        pastoTest.aggiungiRicetta(new Ricetta("Pasta al pesto", "Una deliziosa pasta al pesto", edit, foto, 20, 2, ingredienti));
        calendario.aggiungiPasto(pastoTest);
        Assert.assertEquals(true, calendario.eliminaPasto(pastoTest));

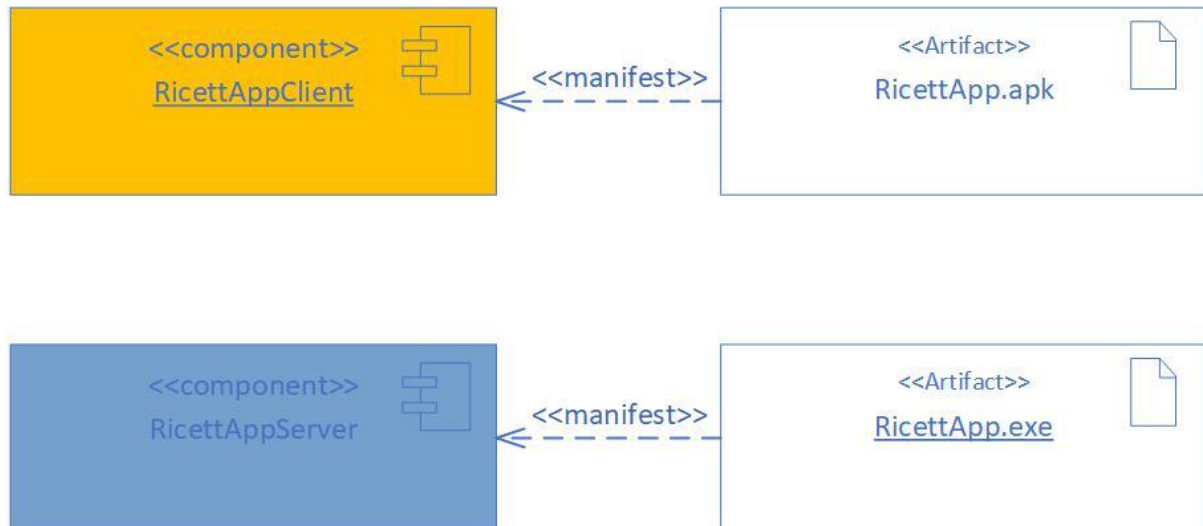
        edit = LocalDate.now();
        foto = new File[10];
        foto[0] = new File("fotoPastaPesto1.jpg");
        ingredienti = new IngredienteConQuantità[2];
        ingredienti[0] = new IngredienteConQuantità(20, ListaDegliIngredienti.getIngrediente("Pasta"), UnitàDiMisura.GRAMMI, 200);
        ingredienti[1] = new IngredienteConQuantità(35, ListaDegliIngredienti.getIngrediente("Pesto"), UnitàDiMisura.GRAMMI, 50);
        pastoTest.eliminaRicetta(new Ricetta("Pasta al pesto", "Una deliziosa pasta al pesto", edit, foto, 20, 2, ingredienti));
        Assert.assertEquals(pasto.getRicette(), new ArrayList<Ricetta>());
    }
}
```

Deployment

Come strumento per il deployment del client si è deciso di sfruttare il Play Store di Google, che semplifica nettamente il download, l'installazione e soprattutto la gestione degli aggiornamenti.

Il server non sarà distribuito, e sarà lanciato su una macchina dedicata.

Artefatti



Deployment type-level

