



Ubp3

Una bottiglia per tre



Premessa

Il documento rappresenta una descrizione relativa allo sviluppo del progetto

Ubp3: Una bottiglia per tre. Il software è stato implementato nel linguaggio Java usando l'ambiente IntelliJ, mentre la progettazione in UML è stata realizzata usando il software Astah. I dati sono mantenuti e gestiti tramite file.

In questo documento è presente solo la versione finale ottenuta al termine delle fasi di progettazione; le versioni intermedie sono nelle relative sottocartelle.

Indice:

1	Ideazione e analisi dei requisiti	4
1.1	Introduzione.....	4
1.2	Requisiti	4
1.3	Obiettivi e casi d'uso.....	5
1.4	Modello dei casi d'uso	6
	UC1: Gestisci Vino, CRUD	6
	UC2: Crea Ordine Fornitore.....	7
	UC3: Gestisci Cliente, CRUD	9
	UC4: Gestisci Vendita.....	10
	UC5: Ricerca Vendita.....	12
	UC6: Gestisci Prenotazione	13
	UC7: Accumula Punti.....	13
	UC8: Gestisci Sconto, CRUD.....	14
	UC9: Rimuovi Punti	16
1.5	Glossario	17
2	Analisi Orientata agli oggetti.....	18
2.1	Introduzione.....	18
2.2	Modello di Dominio.....	19
2.3	SSD e Contratti delle operazioni.....	20
3	Progettazione	34
3.1	Diagramma delle classi	34
3.2	Diagrammi di sequenza	35
4	Testing.....	47
4.1	Introduzione.....	47
4.2	Introduzione dei casi di test e Testing Unitario.....	47

1 Ideazione e analisi dei requisiti

1.1 Introduzione

Lo scopo della fase di ideazione è presentare un quadro generale del progetto con l'obiettivo di sintetizzare rapidamente le idee principali descrivendone i requisiti funzionali e non.

Inoltre, analizza la fattibilità tenendo conto dei tempi e delle risorse necessarie. In questo documento verranno presentati: i requisiti, i casi d'uso e il glossario.

1.2 Requisiti

Il titolare di un'enoteca richiede la realizzazione di un software che permetta di migliorare la gestione e la vendita di vini.

Il software segue il processo di vendita in tutte le sue fasi, a partire dall'inserimento dei vini nel catalogo fino alla vendita degli stessi ai clienti.

In particolare, il titolare deve:

- poter aggiungere, eliminare o modificare i vini presenti nel catalogo;
- effettuare degli ordini alle aziende fornitrici includendo le richieste dei clienti ma anche includendoli arbitrariamente;
- deve poter creare delle promozioni: i clienti potranno accumulare punti in base alla quantità di vini acquistati e sulla base di questi potranno ricevere un omaggio;
- possibilità di applicare un codice sconto in fase di pagamento;
- tener traccia dei dati dei clienti.

Il cliente:

- comunica al titolare i vini che intende acquistare o la loro tipologia. Il titolare ricercherà nel sistema l'effettiva esistenza del vino e la sua disponibilità in magazzino. Il risultato della ricerca verrà comunicata al cliente:
 1. Se il vino è presente, il cliente potrà acquistarlo o meno.
 1. Se il vino non è in magazzino, il cliente potrà decidere se prenotarlo. In questo caso il titolare memorizzerà i vini che il cliente vorrà prenotare alle aziende fornitrici.
- Se il cliente decide di acquistare, prima di procedere, il titolare chiederà al cliente se vuole registrarsi al software o se è già stata effettuata una registrazione in precedenza. La registrazione sarà utile per poter accumulare punti ed usufruire delle promozioni.

- Nel caso in cui il vino richiesto dal cliente è presente e decide di acquistarlo verrà creata una scheda di acquisto. Il sistema dovrà aggiornare la quantità di vini presenti nel magazzino. Infine, il sistema calcolerà l'importo da pagare; nel caso in cui viene inserito un codice sconto valido il sistema calcolerà il nuovo prezzo di vendita.
- Il titolare potrà aggiungere dei punti al cliente per ogni bottiglia di vino acquistata, utili per ricevere un regalo.
- Se il vino non è in magazzino e il cliente deciderà di prenotarlo, rimarrà in attesa di una comunicazione, tramite chiamata o sms, da parte del titolare.

1.3 Obiettivi e casi d'uso

Tramite i requisiti analizzati precedentemente è stato possibile ricavare i casi d'uso.

Attore	Obiettivo	Caso d'uso
Titolare	Inserire, rimuovere, modificare e ricercare un vino.	UC1: Gestisci Vino (CRUD, <i>Inserisci Vino</i>)
Titolare	Creare un ordine verso un'azienda fornitrice per ordinare i vini richiesti o mancanti.	UC2: Crea Ordine Fornitore
Titolare	Ricerca, inserire, rimuovere o modificare i dati del cliente.	UC3: Gestisci Cliente (CRUD, <i>Inserisci Cliente</i>)
Titolare	Gestire la vendita di vini considerando eventuali sconti.	UC4: Gestisci Vendita
Titolare	Ricerca una determinata vendita.	UC5: Ricerca Vendita
Titolare	Crea una lista di vini prenotati dai clienti.	UC6: Gestisci Prenotazione
Titolare	Aggiungere i punti ad un relativo cliente.	UC7: Accumula Punti
Titolare	Creare, modificare o rimuovere sconti.	UC8: Gestisci Sconto (CRUD, <i>Crea Sconto</i>)
Titolare	Sottrarre i punti ad un relativo cliente.	UC9: Rimuovi Punti

1.4 Modello dei casi d'uso

UC1: Gestisci Vino, CRUD

Nome del caso d'uso	UC1: Gestisci Vino
Portata	Ubp3
Livello	Obiettivo utente, CRUD
Attore primario	Titolare
Parti interessate e interessi	<ul style="list-style-type: none">▪ Titolare: vuole gestire cioè creare, cercare, modificare, cancellare e visualizzare i vini dalla lista in maniera precisa e corretta.
Pre-condizioni	
Garanzia di successo	Consistenza dei dati.
Scenario principale di successo	<ol style="list-style-type: none">1. Il titolare chiede al sistema l'inserimento di un nuovo vino.2. Il sistema chiede al titolare le informazioni relative al vino.3. Il titolare inserisce nomeAzienda, partitalva, IDvino, tipologia, marchio, provenienza, prezzo e quantitàDisponibili.4. Il sistema cerca se il vino è già presente in magazzino e in caso contrario lo crea.5. Il sistema stampa a video il vino appena inserito e chiede al titolare di confermare l'inserimento.6. Il titolare conferma.7. Il sistema aggiunge il nuovo vino in lista.
Estensioni	<ul style="list-style-type: none">• In qualsiasi momento il sistema potrebbe arrestarsi.<ol style="list-style-type: none">1. Il titolare riavvia il software e ripristina lo stato precedente del sistema.2. Il sistema ricrea lo stato precedente.1.1 Il titolare chiede al sistema la modifica di un vino in lista.<ol style="list-style-type: none">1. Il sistema chiede al titolare i dati aggiornati.2. Il titolare aggiorna i campi di interesse (nomeAzienda, prtitalva, IDvino, tipologia, marchio, provenienza, prezzo e quantitàDisponibili).3. Il sistema stampa a video il vino appena modificato e chiede al titolare se vuole confermare la modifica.4. Il titolare conferma.5. Il vino viene modificato.1.2 Il titolare chiede al sistema di ricercare un vino in lista.<ol style="list-style-type: none">1. Il sistema chiede al titolare di inserire le informazioni.2. Il titolare inserisce le informazioni a sua

	<p>disposizione.</p> <p>3. Il sistema mostra l'elenco dei vini che soddisfano i criteri di ricerca.</p> <p>1.3 Il titolare chiede al sistema di eliminare un vino dal catalogo.</p> <ol style="list-style-type: none"> 1. Il sistema chiede al titolare di inserire le informazioni del vino da eliminare. 2. Il titolare inserisce i dati. 3. Il sistema mostra a video il vino e chiede al titolare di confermare l'eliminazione. 4. Il titolare conferma. 5. Il sistema elimina il vino dalla lista.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	Non specificati.
Frequenza di ripetizione	La gestione dei vini è un'operazione che viene svolta molto frequentemente dal titolare.
Problemi aperti	Non specificati.

UC2: Crea Ordine Fornitore

Nome del caso d'uso	UC2: Crea Ordine Fornitore
Portata	Ubp3.
Livello	Obiettivo utente.
Attore primario	Titolare.
Parti interessate e interessi	<p>Titolare: controlla le prenotazioni relative ai clienti per una determinata azienda e inoltre inserisce dei vini a sua scelta. Nel momento della creazione dell'ordine l'attore primario desidera inserire in maniera precisa e senza errori i dati.</p> <p>Azienda fornitrice: desidera ricevere gli ordini in maniera precisa e chiara.</p>
Pre-condizioni	I precedenti ordini sono già stati consegnati alle aziende fornitrici.
Garanzia di successo	Viene creato un documento di testo con i dati relativi dell'ordine che sarà poi inoltrato all'azienda.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il titolare vuole effettuare un nuovo ordine per un'azienda. 2. Il sistema chiede al titolare l'inserimento del nome e

	<p>della partita iva dell'azienda.</p> <ol style="list-style-type: none"> 3. Il titolare inserisce i dati. 4. Ubp3 mostra a video la lista dei vini prenotati dai clienti per quella determinata azienda e li include nella lista dei vini ordinati. 5. Il sistema elimina dalla lista prenotazioni cliente tutti i vini prenotati per quella determinata azienda. 6. Il sistema chiede l'inserimento dei dati del vino che il titolare vuole ordinare. 7. Il titolare inserisce i dati. 8. Il sistema aggiunge il vino alla lista e chiede al titolare se ha concluso l'ordine o intende ancora prenotare. 9. Ripetere i passi cinque, sei e sette per ogni tipologia di vino che si intende ordinare. 10. Il sistema chiede al titolare se vuole confermare la creazione di un nuovo ordine. 11. Il titolare conferma l'ordine. 12. Il sistema crea un file di testo contenente l'ordine. 13. Il titolare a questo punto potrà inviare il file all'azienda fornitrice.
Estensioni	<ul style="list-style-type: none"> • In qualsiasi momento il sistema potrebbe arrestarsi. <ol style="list-style-type: none"> 1. Il titolare riavvia il software e ripristina lo stato precedente del sistema. 2. Il sistema ricrea lo stato precedente. 10.1. Il titolare vuole annullare l'ordine. <ol style="list-style-type: none"> 1. Il titolare chiede al sistema di annullare questa operazione. 2. Il sistema conferma l'annullamento e ripristina la lista prenotazioni clienti.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	Non specificati.
Frequenza di ripetizione	In base alle richieste del titolare e dei clienti.
Problemi aperti	Non specificati.

UC3: Gestisci Cliente, CRUD

Nome del caso d'uso	UC3: Gestisci Cliente
Portata	Ubp3
Livello	Obiettivo utente, CRUD
Attore primario	Titolare
Parti interessate e interessi	Titolare: vuole gestire cioè inserire, cercare, modificare, cancellare e visualizzare i clienti in maniera semplice e corretta.
Pre-condizioni	
Garanzia di successo	Consistenza dei dati.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il titolare chiede al sistema l'inserimento di un nuovo cliente. 2. Il sistema chiede al titolare le informazioni relative al cliente. 3. Il titolare inserisce nome, cognome, email e numeroTelefono. 4. Il sistema cerca se il cliente è già presente in lista e in caso contrario lo crea. 5. Il sistema stampa a video il cliente appena inserito e chiede al titolare di confermare l'inserimento. 6. Il titolare conferma l'inserimento. 7. Il sistema aggiunge il nuovo cliente in lista.
Estensioni	<ul style="list-style-type: none"> ▪ In qualsiasi momento, il sistema fallisce e ha un arresto improvviso. <ol style="list-style-type: none"> 1. Il titolare riavvia il software e richiede il ripristino dello stato precedente del sistema. 2. Il sistema ricostruisce lo stato precedente. 1.1 Il titolare chiede al sistema la modifica di un cliente. <ol style="list-style-type: none"> 1. Il sistema chiede al titolare i dati che devono essere aggiornati. 2. Il titolare aggiorna i campi di interesse. 3. Il sistema mostra a video il cliente e chiede di confermare la modifica. 4. Il titolare conferma. 1.2 Il titolare chiede al sistema di ricercare un cliente. <ol style="list-style-type: none"> 1. Il sistema chiede al titolare di inserire le informazioni. 2. Il titolare inserisce le informazioni.

	<ol style="list-style-type: none"> 3. Il sistema mostra l'elenco dei clienti che soddisfano i criteri di ricerca. <p>1.3 Il titolare chiede al sistema di eliminare un cliente</p> <ol style="list-style-type: none"> 1. Il sistema chiede al titolare di inserire le informazioni del cliente da eliminare. 2. Il titolare inserisce i dati. 3. Il sistema mostra a video il cliente e chiede al titolare di confermare l'eliminazione. 4. Il titolare conferma. 5. Il sistema elimina il cliente dalla lista.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	Non specificati.
Frequenza di ripetizione	Dipende dal numero di acquisti e di prenotazioni presenti.
Problemi aperti	Non specificati.

UC4: Gestisci Vendita

Nome del caso d'uso	UC4: Gestisci Vendita
Portata	Ubp3.
Livello	Obiettivo utente.
Attore primario	Titolare.
Parti interessate e interessi	<p>Titolare: vuole gestire il processo di vendita in modo semplice, funzionale e veloce.</p> <p>Cliente: desidera acquistare i vini in maniera semplice e veloce.</p>
Pre-condizioni	Il cliente sa quali sono i vini che vuole acquistare o l'azienda a cui appartengono.
Garanzia di successo	<p>La vendita si conclude con il pagamento e la consegna del vino al cliente.</p> <p>Inoltre, vengono aggiornate le quantità presenti in magazzino.</p>
Scenario principale di successo	<ol style="list-style-type: none"> 1. Un cliente si reca in enoteca e chiede al titolare un vino da acquistare. 2. Il titolare utilizza il software Ubp3 per cercare nel catalogo il vino richiesto inserendo il marchio e la tipologia. Il sistema mostra al titolare il vino trovato e le quantità presenti in magazzino. Il titolare lo comunica al cliente. 3. Il cliente vuole acquistare il vino e comunica le quantità di vino da acquistare. 4. Il titolare utilizza Ubp3 per inserire le quantità ed aggiungere il vino richiesto alla scheda di acquisto.

	<ol style="list-style-type: none"> 5. Il sistema aggiorna le quantità presenti in magazzino. 6. Ripetere i passi due, tre, quattro e cinque per ogni tipologia di vino richiesto dal cliente. 7. Il cliente non vuole aggiungere altri vini alla scheda di acquisto. 8. Il sistema calcola il prezzo totale dell'ordine e chiede al titolare di inserire l'eventuale codice sconto. 9. Il titolare inserisce il codice sconto. 10. Il sistema applica la percentuale di sconto e calcola il prezzo da pagare. 11. Il cliente fornisce i propri dati al titolare. 12. Il titolare digita il nome, il cognome e l'email del cliente e lo associa alla scheda di acquisto. 13. Il titolare conferma l'operazione. 14. Il cliente si reca alla cassa, paga e ritira l'acquisto.
Estensioni	<ul style="list-style-type: none"> ▪ In qualsiasi momento, il sistema fallisce e ha un arresto improvviso. <ol style="list-style-type: none"> 1. Il titolare riavvia il software e richiede il ripristino dello stato precedente del sistema. 2. Il sistema ricostruisce lo stato precedente. 1.1 Il cliente indica al titolare la tipologia di vino da acquistare. <ol style="list-style-type: none"> 1. Il titolare effettua la ricerca <i>[vedi caso d'uso Gestisci Vino]</i>. 2. Il sistema restituisce l'elenco dei vini disponibili. 2.1 Il vino non è presente nel catalogo. <ol style="list-style-type: none"> 1. Il titolare lo comunica al cliente. 3.1 Il cliente non vuole acquistare il vino. 4.1 Le quantità richieste dal cliente non sono presenti in magazzino. <ol style="list-style-type: none"> 1. Il vino non viene inserito nella scheda di acquisto. 9.1 Il titolare non inserisce il codice sconto. 10.1 Il codice inserito non è valido. 11.1 Il cliente non vuole registrarsi. <ol style="list-style-type: none"> 1. Il cliente acquista senza essersi registrato. 12.1 Il cliente non è presente in archivio. <ol style="list-style-type: none"> 1. Il titolare inserisce i dati nel sistema e conferma <i>[vedi caso d'uso Gestisci Cliente]</i>. 2. Il sistema associa il nuovo cliente alla vendita in corso. 13.1 Il cliente non vuole proseguire con l'acquisto e va via. <ol style="list-style-type: none"> 1. Il titolare annulla la vendita appena creata. 2. Il sistema ripristina le quantità dei vini in

	magazzino.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	Non specificati.
Frequenza di ripetizione	In base alle richieste dei clienti e agli acquisti che vogliono effettuare.
Varie	Non specificati.

UC5: Ricerca Vendita

Nome del caso d'uso	UC5: Ricerca Vendita
Portata	Ubp3.
Livello	Obiettivo utente.
Attore primario	Titolare.
Parti interessate e interessi	Titolare: vuole ricercare una vendita in modo semplice, funzionale e veloce.
Pre-condizioni	
Garanzia di successo	Consistenza dei dati in output.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il titolare chiede al sistema di cercare delle vendite esistenti. 2. Il sistema chiede la data delle vendite da cercare. 3. Il titolare inserisce la data e conferma. 4. Il sistema restituisce le vendite corrispondenti
Estensioni	<ul style="list-style-type: none"> ▪ In qualsiasi momento, il sistema fallisce e ha un arresto improvviso. <ol style="list-style-type: none"> 1. Il titolare riavvia il software e richiede il ripristino dello stato precedente del sistema. 2. Il sistema ricostruisce lo stato precedente. <p>4.1 Non esiste nessuna vendita effettuata in quella data.</p>
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	Non specificati.
Frequenza di ripetizione	In base alle necessità del titolare.
Varie	Non specificati.

UC6: Gestisci Prenotazione

Nome del caso d'uso	UC6: Gestisci Prenotazione
Portata	Ubp3.
Livello	Obiettivo utente.
Attore primario	Titolare.
Parti interessate e interessi	Cliente: vuole prenotare dei vini; Titolare: memorizza i vini prenotati dal cliente.
Pre-condizioni	
Garanzia di successo	Il sistema registra in un file di testo le prenotazioni e i relativi clienti.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il titolare vuole memorizzare dei vini prenotati da un cliente. 2. Il sistema chiede al titolare di inserire i dati relativi ad ogni singolo vino e le rispettive quantità prenotate. 3. Il titolare inserisce i dati. 4. Ripetere i passi 2 e 3 per ogni vino prenotato. 5. Il sistema chiede al titolare di inserire le informazioni del cliente. 5. Il titolare inserisce i dati del cliente. 6. Il sistema creerà un file di testo contenente i vini prenotati associati ai rispettivi clienti (utile per il titolare).
Estensioni	<ul style="list-style-type: none"> ▪ In qualsiasi momento, il sistema fallisce e ha un arresto improvviso. <ol style="list-style-type: none"> 1. Il titolare riavvia il software e richiede il ripristino dello stato precedente del sistema. 2. Il sistema ricostruisce lo stato precedente.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	Non specificati.
Frequenza di ripetizione	Dipende dalle prenotazioni dei clienti.
Varie	Non specificati.

UC7: Accumula Punti

Nome del caso d'uso	UC7: Accumula Punti
Portata	Ubp3.
Livello	Obiettivo utente.
Attore primario	Titolare.
Parti interessate e interessi	Titolare: vuole inserire dei punti ad un cliente.
Pre-condizioni	

Garanzia di successo	Consistenza dei dati.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il titolare vuole aggiungere dei punti ad un relativo cliente. 2. Il sistema chiede al titolare di fornire i dati del cliente. 3. Il titolare inserisce le informazioni relative al cliente. 4. Il sistema ricerca il cliente e chiede i punti da aggiungere. 5. Il titolare inserisce i punti. 6. Il sistema aggiunge i punti e mostra a video le informazioni del cliente con i relativi punti aggiornati.
Estensioni	<ul style="list-style-type: none"> ▪ In qualsiasi momento, il sistema fallisce e ha un arresto improvviso. <ol style="list-style-type: none"> 1. Il titolare riavvia il software e richiede il ripristino dello stato precedente del sistema. 2. Il sistema ricostruisce lo stato precedente. <p>4.1 Il cliente non è presente nella lista clienti.</p>
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	Non specificati.
Frequenza di ripetizione	In base alla frequenza degli acquisti di un dato cliente.
Varie	Non specificati.

UC8: Gestisci Sconto, CRUD

Nome del caso d'uso	UC8: Gestisci Sconto
Portata	Ubp3
Livello	Obiettivo utente, CRUD
Attore primario	Titolare
Parti interessate e interessi	Titolare: vuole gestire cioè inserire, cercare, modificare, cancellare e visualizzare i codici sconto in maniera semplice e corretta.
Pre-condizioni	
Garanzia di successo	Consistenza dei dati.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il titolare chiede al sistema l'inserimento di un nuovo codice sconto. 2. Il sistema chiede al titolare le informazioni relative al codice sconto. 3. Il titolare inserisce codiceSconto e percentuale.

	<ol style="list-style-type: none"> 4. Il sistema cerca se il codice sconto è già presente in lista e in caso contrario lo crea. 5. Il sistema stampa a video il codice sconto appena inserito e chiede al titolare di confermare l'inserimento. 6. Il titolare conferma l'inserimento. 7. Il sistema aggiunge il nuovo codice sconto in lista.
Estensioni	<ul style="list-style-type: none"> ▪ In qualsiasi momento, il sistema fallisce e ha un arresto improvviso. <ol style="list-style-type: none"> 1. Il titolare riavvia il software e richiede il ripristino dello stato precedente del sistema. 2. Il sistema ricostruisce lo stato precedente. 1.1 Il titolare chiede al sistema la modifica di un codice sconto. <ol style="list-style-type: none"> 1. Il sistema chiede al titolare i dati che devono essere aggiornati. 2. Il titolare aggiorna i campi di interesse. 3. Il sistema mostra a video il codice sconto e chiede di confermare la modifica. 4. Il titolare conferma. 1.2 Il titolare chiede al sistema di ricercare un codice sconto. <ol style="list-style-type: none"> 1. Il sistema chiede al titolare di inserire le informazioni. 2. Il titolare inserisce le informazioni. 3. Il sistema mostra l'elenco dei codici sconto che soddisfano i criteri di ricerca. 1.3 Il titolare chiede al sistema di eliminare un codice sconto <ol style="list-style-type: none"> 1. Il sistema chiede al titolare di inserire le informazioni del codice sconto da eliminare. 2. Il titolare inserisce i dati. 3. Il sistema mostra a video il codice sconto e chiede al titolare di confermare l'eliminazione. 4. Il titolare conferma. 5. Il sistema elimina il codice sconto dalla lista.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	Non specificati.
Frequenza di ripetizione	Dipende dal volere del titolare.
Problemi aperti	Non specificati.

UC9: Rimuovi Punti

Nome del caso d'uso	UC9:Rimuovi Punti
Portata	Ubp3.
Livello	Obiettivo utente.
Attore primario	Titolare.
Parti interessate e interessi	Titolare: vuole rimuovere dei punti ad un cliente.
Pre-condizioni	
Garanzia di successo	Consistenza dei dati.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il titolare vuole rimuovere dei punti ad un relativo cliente. 2. Il sistema chiede al titolare di fornire i dati del cliente. 3. Il titolare inserisce le informazioni relative al cliente. 4. Il sistema ricerca il cliente e chiede la quantità di punti da sottrarre. 5. Il titolare inserisce i punti. 6. Il sistema rimuove i punti e mostra a video le informazioni del cliente con i relativi punti aggiornati.
Estensioni	<ul style="list-style-type: none"> ▪ In qualsiasi momento, il sistema fallisce e ha un arresto improvviso. <ol style="list-style-type: none"> 3. Il titolare riavvia il software e richiede il ripristino dello stato precedente del sistema. 4. Il sistema ricostruisce lo stato precedente. 4.1 Il cliente non è presente nella lista clienti.
Requisiti speciali	Non specificati.
Elenco delle varianti tecnologiche e dei dati	Non specificati.
Frequenza di ripetizione	In base alla frequenza degli acquisti di un dato cliente.
Varie	Non specificati.

1.5 Glossario

Qui sono riportati i termini significativi e le loro rispettive definizioni.

Termine	Definizione
Vendita	Processo che inizia con l'arrivo del cliente in enoteca e che termina con il pagamento e il ritiro dell'acquisto.
Acquisto	Azione mediante la quale il cliente entra in possesso dei vini.
Disponibilità	Numero dei pezzi presenti in magazzino.
Azienda	Impresa che si occupa della produzione e distribuzione dei vini.
Ordine	Ordine di uno o più vini da un'azienda fornitrice.
Prenotazione	Prenotazione di una o più tipologie di vini richiesti da un cliente.
Magazzino	Luogo fisico adibito al deposito dei vini.
ID	Sequenza univoca numerica usata per la classificazione e l'identificazione dei vini.
Codice sconto	Stringa di caratteri usata per applicare una determinata percentuale di sconto sul prezzo totale dell'acquisto.
Punti accumulati	Serve per incentivare l'acquisto dei vini con l'opportunità di ricevere un omaggio.

2 Analisi Orientata agli oggetti

2.1 Introduzione

La fase di elaborazione ha l'obiettivo di: raffinare la Visione, implementare il nucleo dell'architettura del software, risolvere le eventuali problematiche collegate ai rischi, individuare i requisiti e fornire stime realistiche del piano di lavoro.

- Iterazione 1:
 - ◆ *UC1: Gestisci Vino*, è un caso d'uso CRUD e si occupa di inserire, modificare, cercare ed eliminare i vini memorizzati nel software. Questi vini rappresentano i vini presenti in magazzino.
 - ◆ *UC2: Crea Ordine Fornitore*, con questo caso d'uso il titolare può effettuare degli ordini dei vini ad una determinata azienda fornitrice. Ubp3 si occuperà di inserire le relative prenotazioni in un file di testo.
 - ◆ *UC3: Gestisce Cliente*, è un caso d'uso CRUD e si occupa di inserire, modificare, cercare ed eliminare i clienti memorizzati nel software.
- Iterazione 2:
 - ◆ *UC4: Gestisci Vendita*, ha l'obiettivo di gestire il processo di vendita dei vini da parte di un cliente in modo semplice e funzionale, memorizzando queste in un file di testo.
 - ◆ *UC5: Ricerca Vendita*, si occupa di ricercare delle vendite attraverso l'inserimento di una determinata data.
 - ◆ *UC6: Gestisci Prenotazione*, permette ai clienti di poter prenotare dei vini.
- Iterazione 3:
 - ◆ *UC4: Gestisci Vendita Estensione 10.1*, viene mostrata la gestione di una vendita in presenza di un codice sconto non valido.
 - ◆ *UC4: Gestisci Vendita Estensione 11.1*, viene effettuata una vendita senza l'associazione di un cliente.
 - ◆ *UC7: Accumula Punti*, si occupa di aggiungere i punti ad un dato cliente.
 - ◆ *UC8: Gestisci Codice Sconto*, è un caso d'uso CRUD che si occupa di inserire, modificare, cercare ed eliminare un codice sconto.
 - ◆ *UC9: Rimuovi Punti*, si occupa di sottrarre i punti ad un dato cliente.

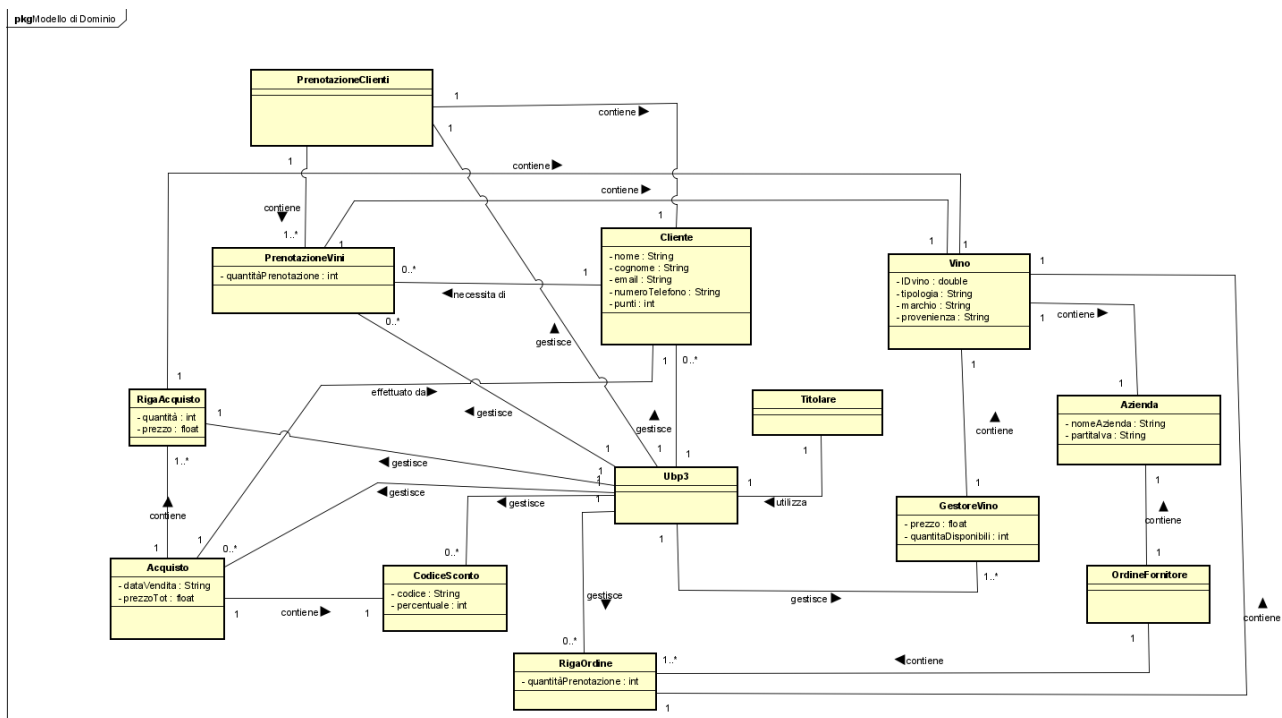
Il passo iniziale di ciascuna iterazione è stato quello di effettuare un'analisi dei requisiti Orientata agli Oggetti, cioè un'analisi basata sulla creazione di una descrizione del dominio da un punto di vista ad oggetti. Per fornire tale descrizione sono stati utilizzati diversi strumenti: Modello di Dominio, SSD (Sequence System Diagram) e i Contratti delle operazioni.

2.2 Modello di Dominio

Sono state identificate le seguenti classi concettuali:

- *Ubp3*: il software “Una bottiglia per tre”.
- *Titolare*: colui che interagisce con il sistema.
- *Vino*: contenente un ID numerico, la tipologia, il marchio e la provenienza del vino.
- *GestoreVino*: contiene il prezzo e le quantità disponibili.
- *Azienda*: contiene il nome dell'azienda e la sua partita iva.
- *OrdineFornitore*: contiene l'ordine da effettuare all'azienda.
- *RigaOrdine*: contiene le quantità dei vini che sono stati ordinati.
- *Cliente*: contiene il nome, il cognome, l'email, il numero di telefono e i punti.
- *PrenotazioneVini*: contiene la quantità dei vini prenotati dai clienti per ogni tipologia.
- *Acquisto*: contiene la data e il prezzo di una particolare vendita e il cliente associato all'acquisto.
- *RigaAcquisto*: contiene le quantità dei vini che sono stati acquistati per ogni tipologia ed il prezzo associato.
- *CodiceSconto*: contiene il codice e la percentuale di uno sconto.
- *PrenotazioniCliente*: contiene la lista dei vini prenotati da un cliente.

Da cui, tenendo conto di associazioni e attributi, è stato ricavato il seguente Modello di Dominio:

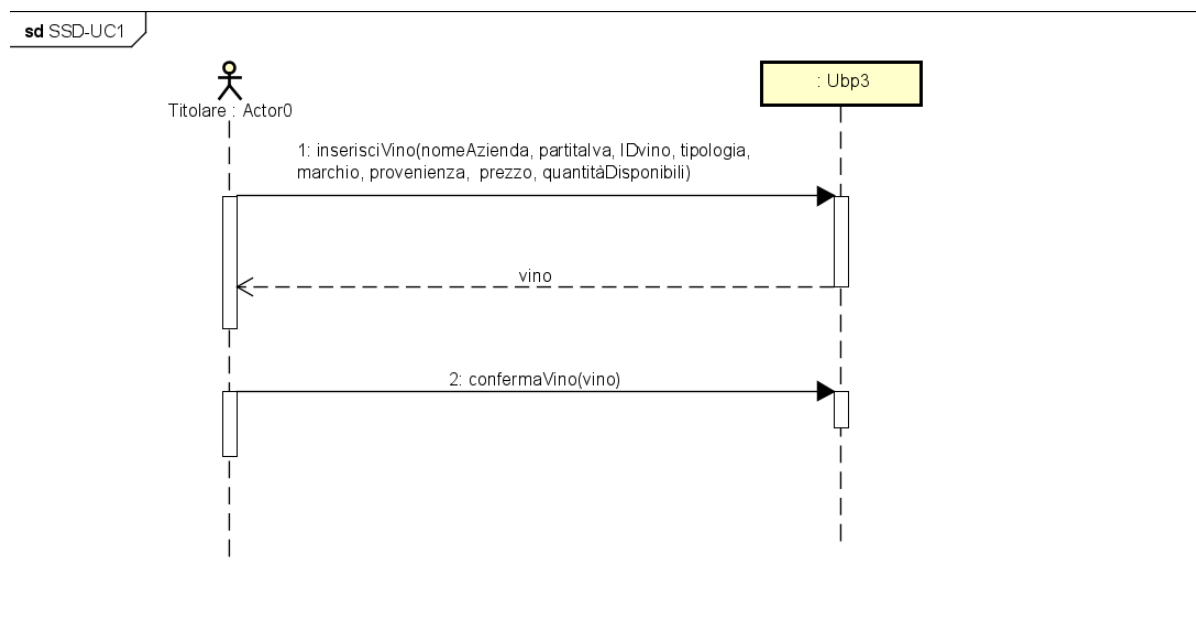


2.3 SSD e Contratti delle operazioni

Procedendo con l'analisi Orientata agli Oggetti, il passo successivo è la creazione del Diagramma di Sequenza di Sistema (SSD) al fine di illustrare il corso degli eventi di input e di output.

Iterazione 1:

UC1:



Contratto CO1: inserisciVino

Operazione: *inserisciVino(nomeAzienda: String, partitalva: String, IDvino: double, tipologia: String, marchio: String, provenienza: String, prezzo: float, quantitàDisponibili: int).*

Riferimenti: caso d'uso: Gestisci Vino.

Pre-condizioni: è in corso l'inserimento di un *vino*.

Post-condizioni: è stata creata un'istanza *vino* di GestoreVino

Contratto CO2: confermaVino

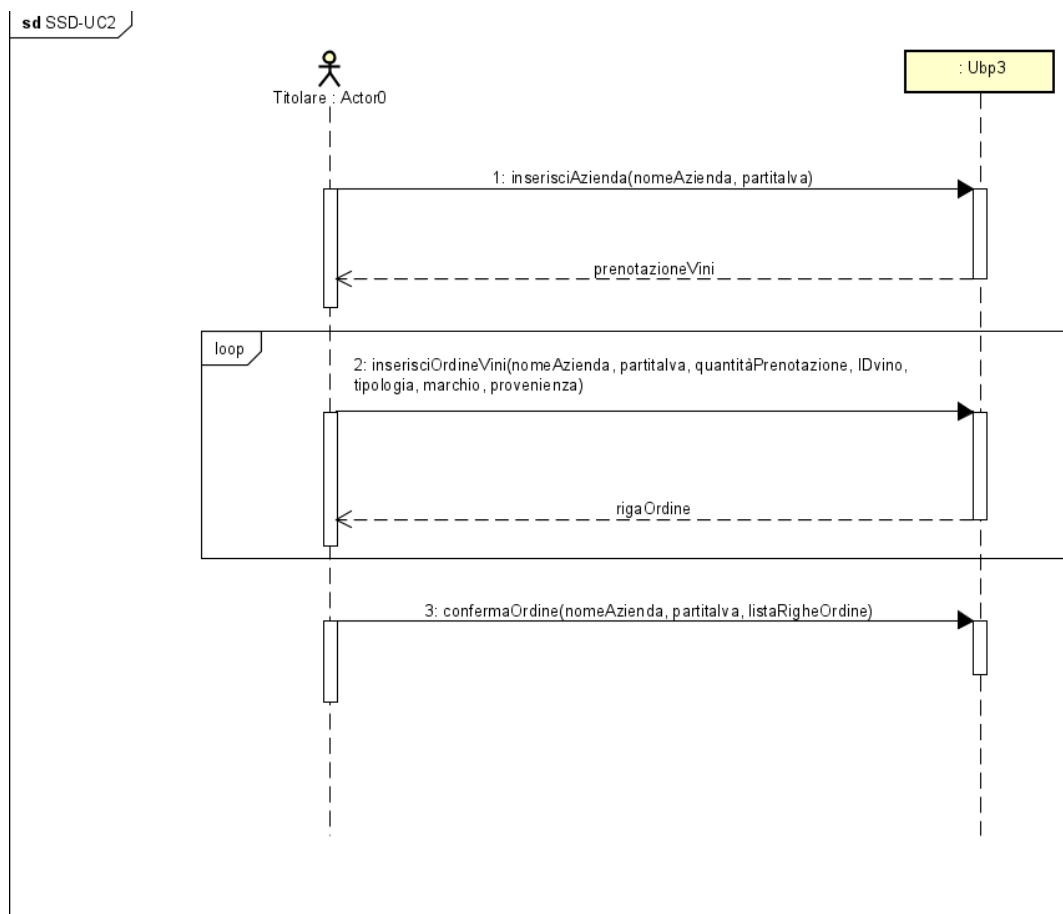
Operazione: *confermaVino(vino: GestoreVino).*

Riferimenti: caso d'uso: Gestisci Vino.

Pre-condizioni: è in corso l'inserimento di un *vino*.

Post-condizioni: - è stato aggiunto un *vino* in lista *vini*(LinkedList <GestoreVino>).
- la lista *vini* è stata aggiornata.

UC2:



Contratto CO1: inserisciAzienda

- Operazione:** *inserisciAzienda(nomeAzienda: String, partitalva: String).*
- Riferimenti:** caso d'uso: Crea Ordine Fornitore.
- Pre-condizioni:** è in corso la creazione di un ordine verso una determinata azienda.
- Post-condizioni:**
- si ricerca all'interno della lista *prenotazioniVini*(LinkedList <*prenotazioneVini*>) i vini prenotati dai clienti per l'azienda specificata.
 - Per ogni prenotazione di *prenotazioniVini*, è stata creata un'istanza *ro* di RigaOrdine.
 - E' stata rimossa dalla lista *prenotazioniVini* la prenotazione effettuata dal cliente.
 - *ro* è stata aggiunta alla lista *prenotazioniiv* (LinkedList <RigaOrdine>).

Contratto CO2: inserisciOrdineVini

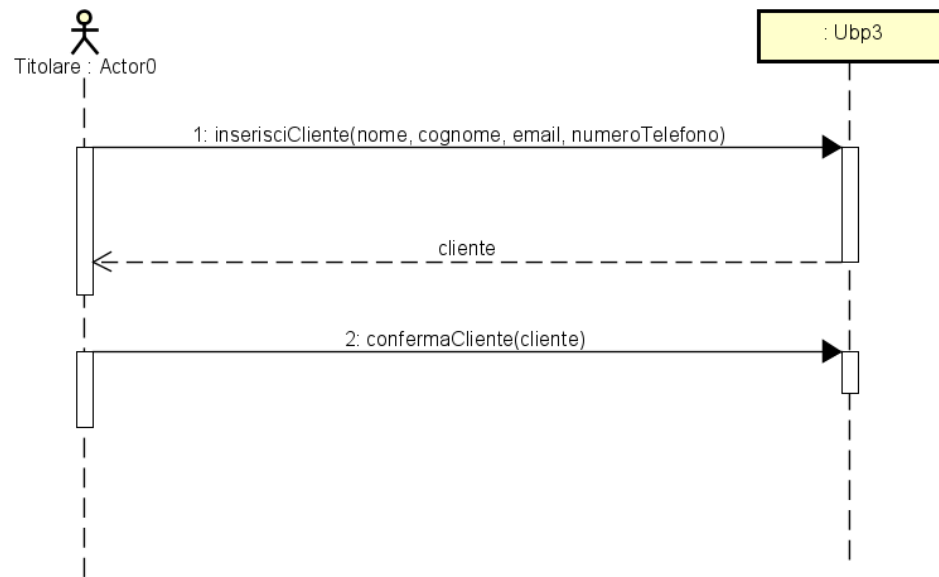
- Operazione:** *inserisciOrdineVini(nomeAzienda: String, partitalva: String, quantitàPrenotazione: int, IDvino: double, tipologia: String, marchio: String, provenienza: String).*
- Riferimenti:** caso d'uso: Crea Ordine Fornitore.
- Pre-condizioni:** è in corso la creazione di un ordine verso una determinata azienda.
- Post-condizioni:** - è stata creata un'istanza *ro* di RigaOrdine.

Contratto CO3: confermaOrdine

- Operazione:** *confermaOrdine(nomeAzienda: String, partitalva: String, listaRigheOrdine LinkedList<RigaOrdine>).*
- Riferimenti:** caso d'uso: Crea Ordine Fornitore.
- Pre-condizioni:** è in corso la creazione di un ordine verso una determinata azienda.
- Post-condizioni:**
- è stata creata un'istanza *ordine* di OrdineFornitore.
 - *ordine* è stato memorizzato all'interno di un file di testo.

UC3:

sd SSD-UC3



Contratto CO1: inserisciCliente

- Operazione:** *inserisciCliente(nome: String, cognome: String, email: String, numeroTelefono: String).*
- Riferimenti:** caso d'uso: Gestisci Cliente.
- Pre-condizioni:** è in corso l'inserimento di un *cliente*.
- Post-condizioni:** è stata creata un'istanza *cliente* di Cliente.

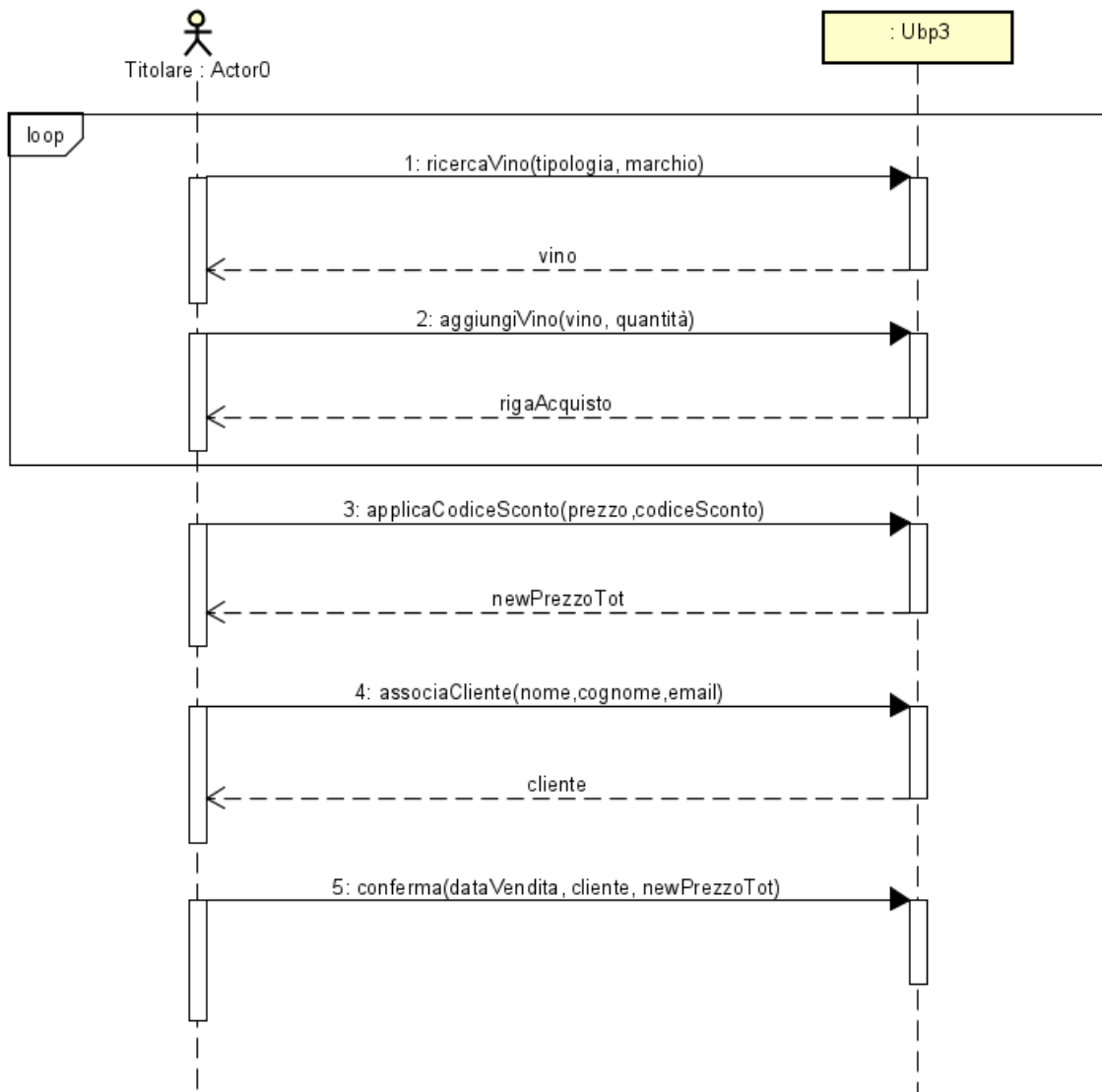
Contratto CO2: confermaCliente

- Operazione:** *confermaCliente(cliente: Cliente).*
- Riferimenti:** caso d'uso: Gestisci Cliente.
- Pre-condizioni:** è in corso l'inserimento di un cliente.
- Post-condizioni:**
- il *cliente* è stato aggiunto alla lista *clienti* (LinkedList <Cliente>).
 - La lista *clienti* è stata aggiornata.

Iterazione 2:

UC4:

sd SSD-UC4



Contratto CO1: ricercaVino

Operazione: *ricercaVino(tipologia: String, marchio: String).*

Riferimenti: caso d'uso: Gestisci Vendita.

Pre-condizioni: è in corso l'acquisto di un *vino*.

Post-condizioni: è stato valutato se il *vino* richiesto è presente in lista.

Contratto CO2: aggiungiVino

Operazione: *aggiungiVino(vino: GestoreVino, quantità: int).*

Riferimenti: caso d'uso: Gestisci Vendita.

- Pre-condizioni:** è in corso l'acquisto di un *vino*.
- Post-condizioni:**
- viene rimosso il *vino* specificato dalla lista *vini*(LinkedList <GestoreVino>).
 - Viene sottratta la quantità specificata al *vino*.
 - Il *vino* modificato viene riaggiunto in lista.
 - Viene creata una nuova *rigaAcquisto*.
 - *rigaAcquisto* è stata aggiunta in lista *righeAcquisto* (LinkedList <RigaAcquisto>).

Contratto CO3: applicaCodisceSconto

- Operazione:** *applicaCodiceSconto*(prezzo: float, codiceSconto: String).
- Riferimenti:** caso d'uso: Gestisci Vendita.
- Pre-condizioni:** è in corso l'acquisto di un vino.
- Post-condizioni:**
- è stato valutato se il *codiceSconto* è valido.
 - E' stata applicata la percentuale di sconto al prezzo.

Contratto CO4: associaCliente

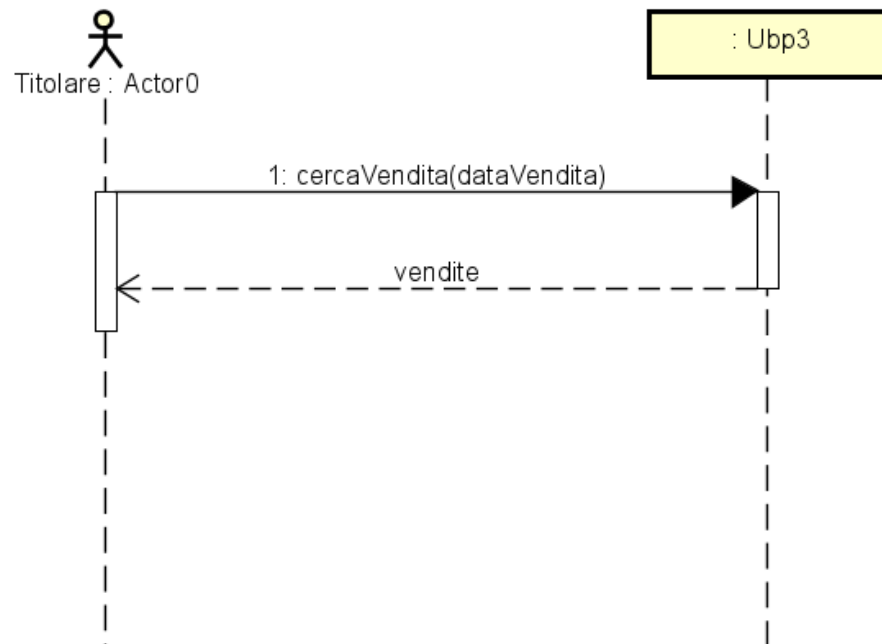
- Operazione:** *associaCliente*(nome: String, cognome: String, email: String).
- Riferimenti:** caso d'uso: Gestisci Vendita.
- Pre-condizioni:** è in corso l'acquisto di un vino.
- Post-condizioni:**
- è stato valutato se il *cliente* è già stato registrato.
 - In caso contrario, viene chiesto al cliente se vuole essere registrato nel software.
 - Viene creata una nuova istanza *cliente*.
 - L'istanza *cliente* viene aggiunta in lista *clienti*(LinkedList <Cliente>).
 - La lista *clienti* viene aggiornata.

Contratto CO5: conferma

- Operazione:** *conferma*(dataVendita: String, cliente: Cliente, newPrezzoTot: float).
- Riferimenti:** caso d'uso: Gestisci Vendita.
- Pre-condizioni:** è in corso l'acquisto di un vino.
- Post-condizioni:**
- è stata creata una nuova istanza *Acquisto*.
 - La nuova istanza è stata memorizzata sul file di testo "Acquisti.txt".

UC5:

sd SSD-UC5



Contratto CO1: cercaVendita

Operazione: *cercaVendita(dataVendita: String).*

Riferimenti: caso d'uso: Ricerca Vendita.

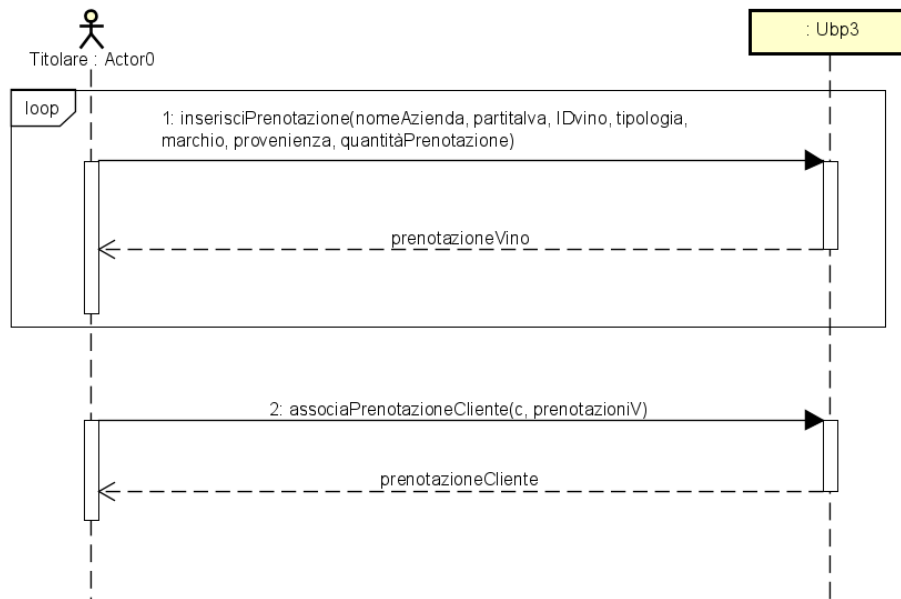
Pre-condizioni: è in corso la ricerca di un acquisto.

Post-condizioni:

- viene copiato il file di testo in una lista *acquisti*(LinkedList <Acquisto>).
- Vengono cercati gli *acquisti* avente quella determinata data.
- Gli *acquisti* vengono inseriti in una lista *vendite*(LinkedList <Acquisto>).
- La lista viene mostrata a video.

UC6:

sd SSD-UC6



Contratto CO1: inserisciPrenotazione

- Operazione:** *inserisciPrenotazione*(nomeAzienda: String, partitalva: String, IDvino: double, tipologia: String, marchio: String, provenienza: String, quatitàPrenotazione: int)
- Riferimenti:** caso d'uso: Gestisci Prenotazione.
- Pre-condizioni:** è in corso la prenotazione di un vino da parte di un cliente.
- Post-condizioni:**
- viene creata una nuova istanza *p* di *prenotazioneVini*.
 - *p* viene aggiunta alla lista *prenotazioni* (LinkedList <PrenotazioneVino>).

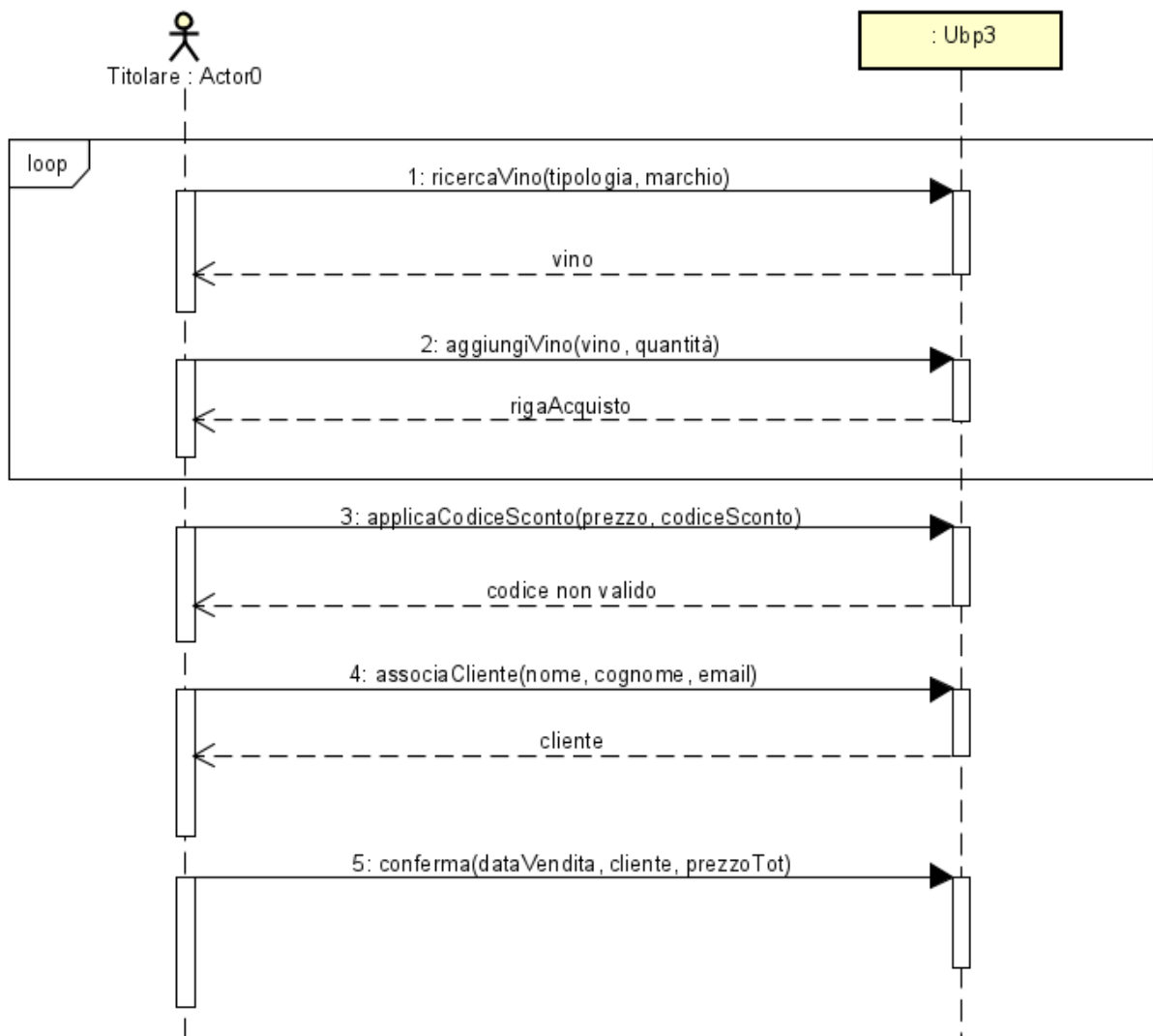
Contratto CO2: associaPrenotazioneCliente

- Operazione:** *associaPrenotazioneCliente*(c : Cliente, prenotazioniV: LinkedList<PrenotazioneVini>).
- Riferimenti:** caso d'uso: Gestisci Prenotazione.
- Pre-condizioni:** è in corso la prenotazione di un vino da parte di un cliente.
- Post-condizioni:**
- è stata creata un'istanza *pc* di PrenotazioneClienti.
 - L'istanza *pc* è stata aggiunta sul file "PrenotazioniClienti.txt".

Iterazione 3:

UC4-Estensione 10.1 :

SD-UC4 Estensione 10.1



Contratto CO1: ricercaVino

Operazione: *ricercaVino(tipologia: String, marchio: String).*

Riferimenti: caso d'uso: Gestisci Vendita.

Pre-condizioni: è in corso l'acquisto di un *vino*.

Post-condizioni: - è stato valutato se il *vino* richiesto è presente in lista.

Contratto CO2: aggiungiVino

Operazione: *aggiungiVino(vino: GestoreVino, quantità: int).*

Riferimenti: caso d'uso: Gestisci Vendita.

Pre-condizioni: è in corso l'acquisto di un *vino*.

- Post-condizioni:**
- viene rimosso il *vino* specificato dalla lista *vini*(LinkedList <GestoreVino>).
 - Viene sottratta la quantità specificata al *vino*.
 - Il *vino* modificato viene riaggiunto in lista.
 - Viene creata una nuova *rigaAcquisto*.
 - *rigaAcquisto* è stata aggiunta in lista *righeAcquisto* (LinkedList <RigaAcquisto>).

Contratto CO3: applicaCodisceSconto

- Operazione:** *applicaCodiceSconto(prezzo: float, codiceSconto: String).*
- Riferimenti:** caso d'uso: Gestisci Vendita.
- Pre-condizioni:** è in corso l'acquisto di un vino.
- Post-condizioni:** - *codiceSconto* non valido.

Contratto CO4: associaCliente

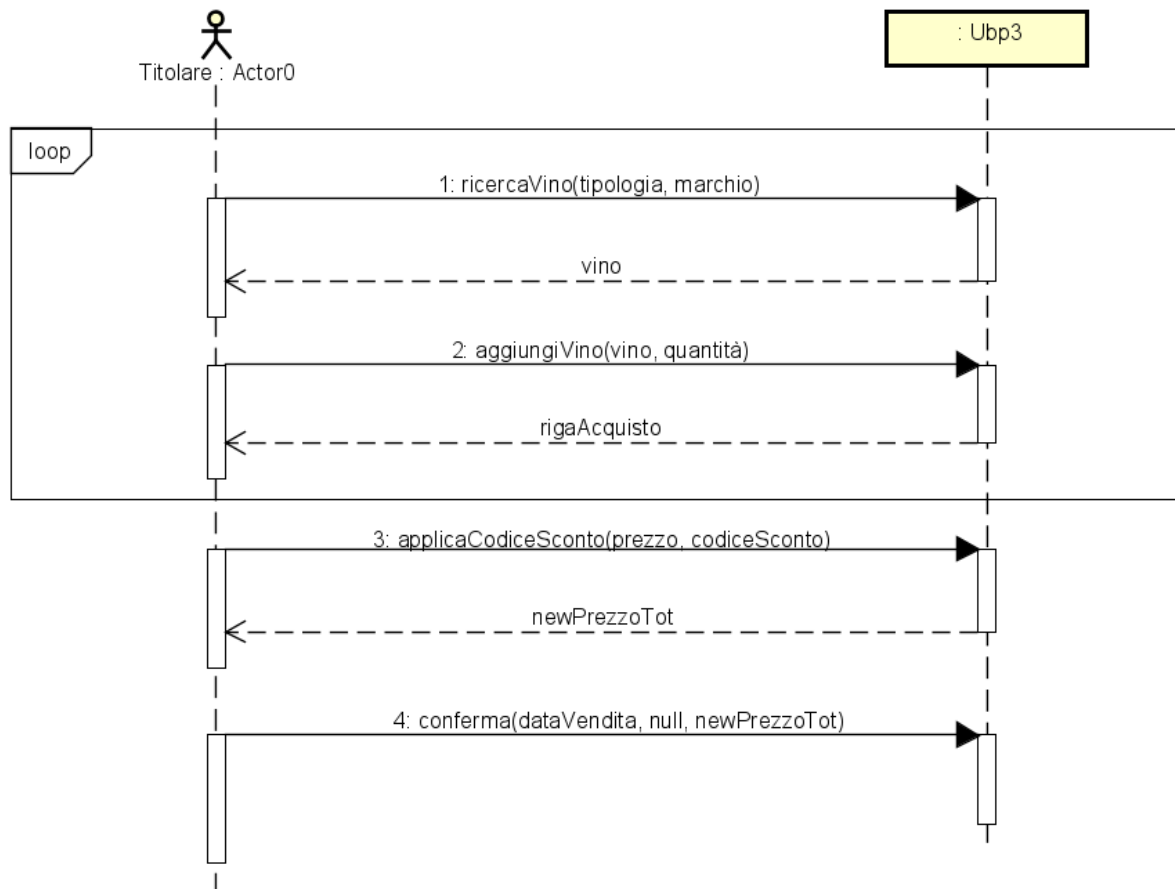
- Operazione:** *associaCliente(nome: String, cognome: String, email: String).*
- Riferimenti:** caso d'uso: Gestisci Vendita.
- Pre-condizioni:** è in corso l'acquisto di un vino.
- Post-condizioni:**
- è stato valutato se il *cliente* è già stato registrato.
 - In caso contrario, viene chiesto al cliente se vuole essere registrato nel software.
 - Viene creata una nuova istanza *cliente*.
 - L'istanza *cliente* viene aggiunta in lista *clienti*(LinkedList <Cliente>).
 - La lista *clienti* viene aggiornata.

Contratto CO5: conferma

- Operazione:** *conferma(dataVendita: String, cliente: Cliente, newPrezzoTot: float).*
- Riferimenti:** caso d'uso: Gestisci Vendita.
- Pre-condizioni:** è in corso l'acquisto di un vino.
- Post-condizioni:**
- è stata creata una nuova istanza *Acquisto*.
 - La nuova istanza è stata memorizzata sul file di testo "Acquisti.txt".

UC4-Estensione 11.1:

sd SSD-UC4 Estensione 11.1



Contratto CO1: ricercaVino

Operazione: *ricercaVino(tipologia: String, marchio: String).*

Riferimenti: caso d'uso: Gestisci Vendita.

Pre-condizioni: è in corso l'acquisto di un *vino*.

Post-condizioni: - è stato valutato se il *vino* richiesto è presente in lista.

Contratto CO2: aggiungiVino

Operazione: *aggiungiVino(vino: GestoreVino, quantità: int).*

Riferimenti: caso d'uso: Gestisci Vendita.

Pre-condizioni: è in corso l'acquisto di un *vino*.

Post-condizioni:

- viene rimosso il *vino* specificato dalla lista *vini*(LinkedList <GestoreVino>).
- Viene sottratta la quantità specificata al *vino*.
- Il *vino* modificato viene riaggiunto in lista.

- Viene creata una nuova *rigaAcquisto*.
- *rigaAcquisto* è stata aggiunta in lista *righeAcquisto* (LinkedList <RigaAcquisto>).

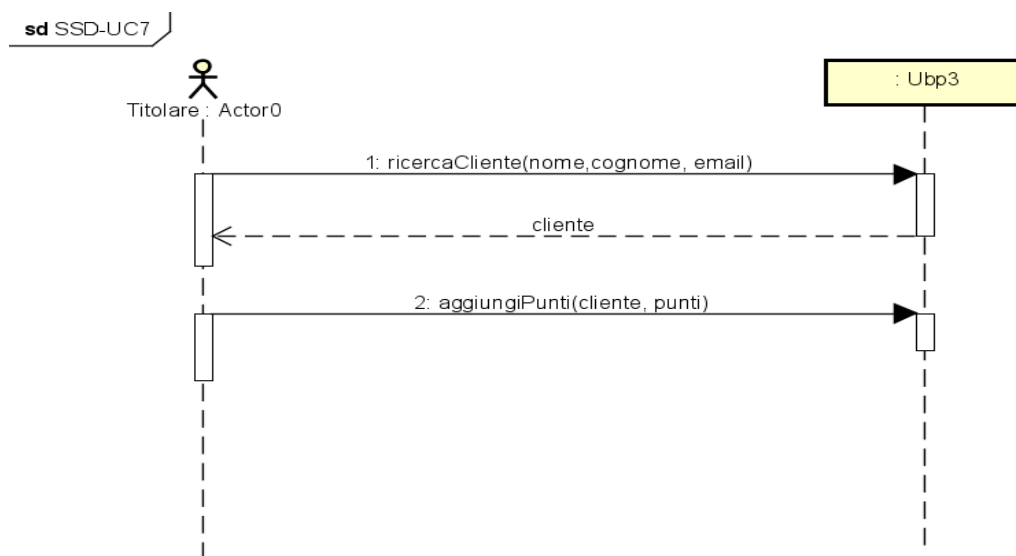
Contratto CO3: applicaCodisceSconto

- Operazione:** *applicaCodiceSconto(prezzo: float, codiceSconto: String).*
- Riferimenti:** caso d'uso: Gestisci Vendita.
- Pre-condizioni:** è in corso l'acquisto di un vino.
- Post-condizioni:** - è stato valutato se il *codiceSconto* è valido.
- E' stata applicata la percentuale di sconto al prezzo.

Contratto CO4: conferma

- Operazione:** *conferma(dataVendita: String, null, newPrezzoTot: float).*
- Riferimenti:** caso d'uso: Gestisci Vendita.
- Pre-condizioni:** è in corso l'acquisto di un vino.
- Post-condizioni:** - è stata creata una nuova istanza *Acquisto*.
- La nuova istanza è stata memorizzata sul file di testo "Acquisti.txt".

UC7:



Contratto CO1: ricercaCliente

Operazione: *ricercaCliente(nome: String, cognome: String, email: String).*

Riferimenti: caso d'uso: Accumula Punti.

Pre-condizioni: è in corso l'aggiunta dei punti ad un cliente.

Post-condizioni: - è stato valutato se il *cliente* è presente in lista.

Contratto CO2: aggiungiPunti

Operazione: *aggiungiPunti(cliente: Cliente, punti: int).*

Riferimenti: caso d'uso: Accumula Punti.

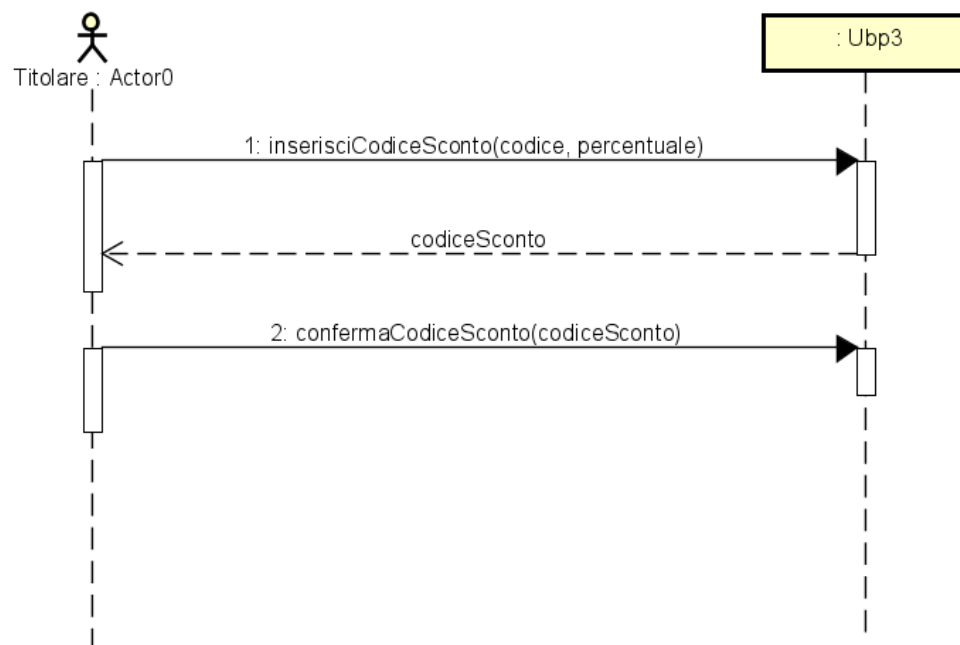
Pre-condizioni: è in corso l'aggiunta dei punti ad un cliente.

Post-condizioni:

- è stato rimosso il *cliente* dalla lista *clienti*(LinkedList <Cliente>).
- Sono stati aggiunti i *punti* al relativo *cliente*.
- Il *cliente* è stato riaggiunto in lista.
- La lista *clienti* è stata aggiornata.

UC8:

sd SSD-UC8



Contratto CO1: inserisciCodiceSconto

Operazione: *inserisciCodiceSconto(codice: String, percentuale: int).*

Riferimenti: caso d'uso: Gestisci Sconto.

Pre-condizioni: è in corso l'inserimento di un codice sconto.

Post-condizioni: è stata creata un'istanza *codiceSconto* di CodiceSconto.

Contratto CO2: confermaCodiceSconto

Operazione: *confermaCodiceSconto(codiceSconto: CodiceSconto).*

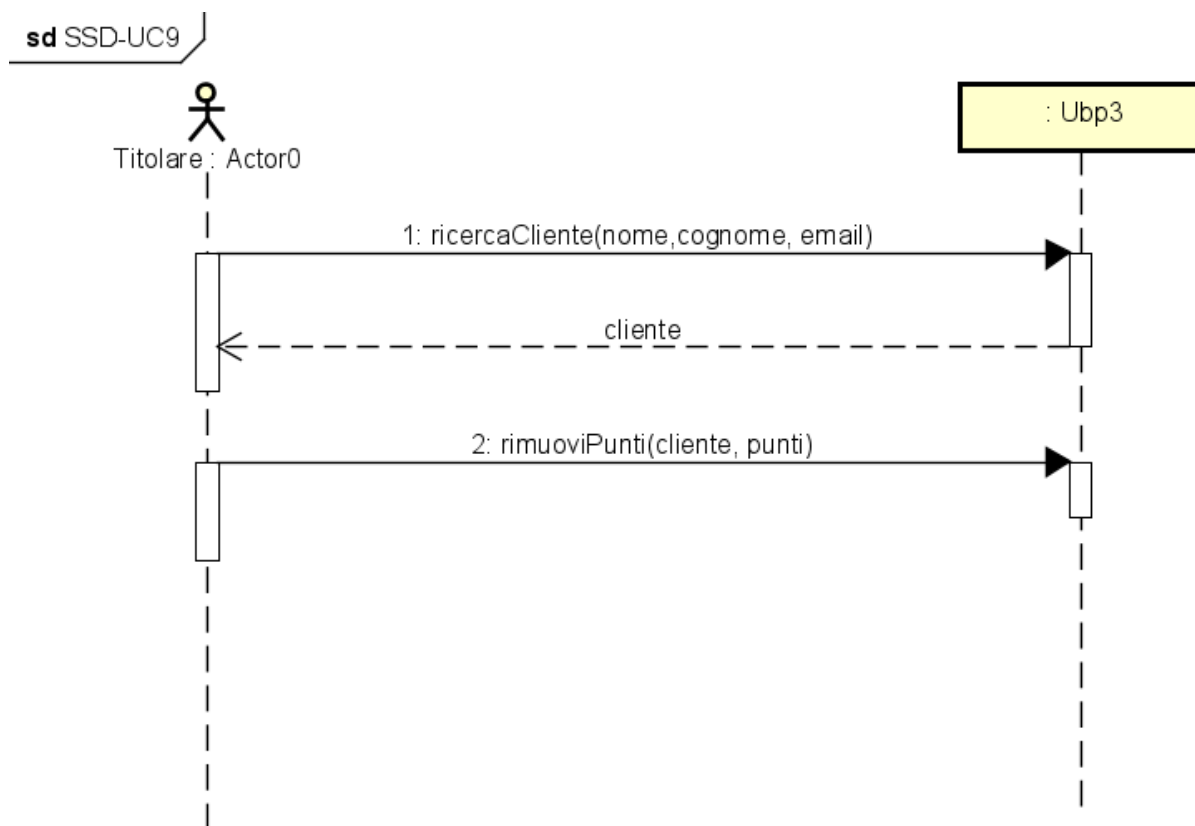
Riferimenti: caso d'uso: Gestisci Sconto.

Pre-condizioni: è in corso l'inserimento di un codice sconto.

Post-condizioni:

- il *codiceSconto* è stato aggiunto alla lista *codiciSconto* (LinkedList <CodiceSconto>).
- La lista *codiciSconto* è stata aggiornata.

UC9:



Contratto CO1: ricercaCliente

Operazione: *ricercaCliente(nome: String, cognome: String, email: String).*

Riferimenti: caso d'uso: Rimuovi Punti.

Pre-condizioni: è in corso la sottrazione dei punti ad un cliente.

Post-condizioni: - è stato valutato se il *cliente* è presente in lista.

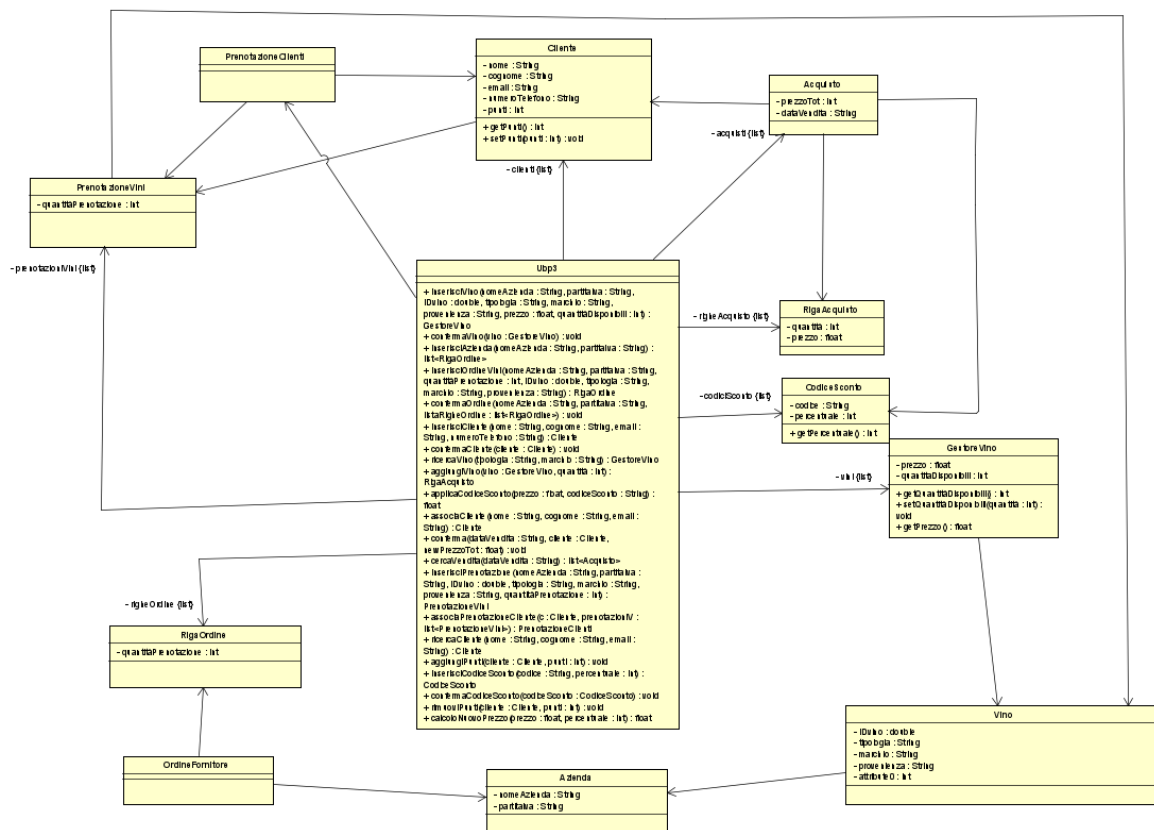
Contratto CO2: rimuoviPunti

- | | |
|-------------------------|---|
| Operazione: | <i>rimuoviPunti(cliente: Cliente, punti: int).</i> |
| Riferimenti: | caso d'uso: Rimuovi Punti. |
| Pre-condizioni: | è in corso la sottrazione dei punti ad un cliente. |
| Post-condizioni: | <ul style="list-style-type: none">- è stato rimosso il <i>cliente</i> dalla lista <i>clienti</i>(LinkedLiskt <Cliente>).- Sono stati sottratti i <i>punti</i> al relativo <i>cliente</i>.- Il <i>cliente</i> è stato riaggiunto in lista.- La lista <i>clienti</i> è stata aggiornata. |

3 Progettazione

3.1 Diagramma delle classi

L'elaborato principale di questa fase è il **Modello di Progetto**, ovvero l'insieme dei diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Interazione) che da un punto di vista statico (Diagramma delle Classi). Il diagramma delle Classi dunque sarà:

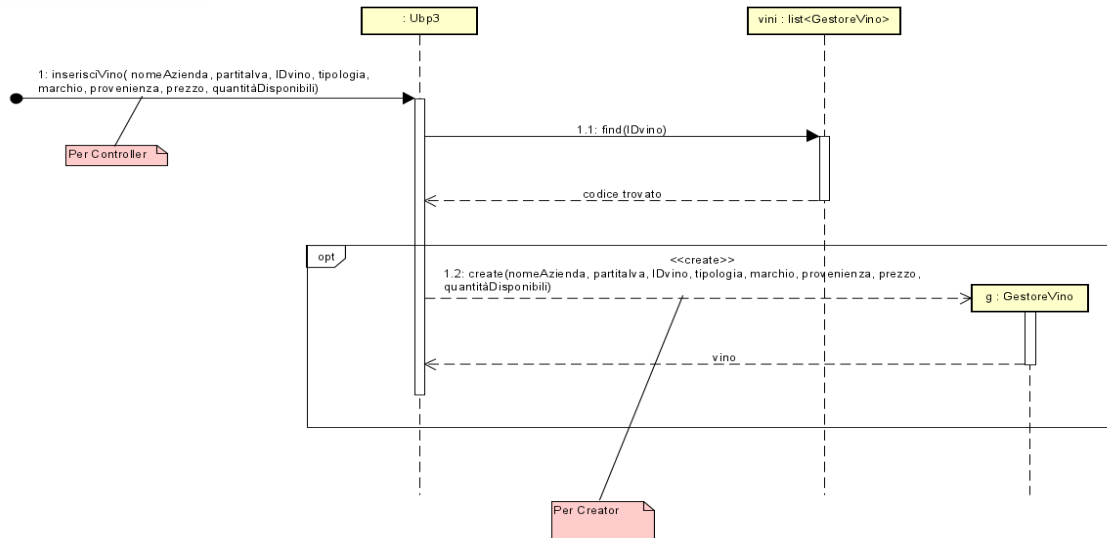


3.2 Diagrammi di sequenza

UC1:

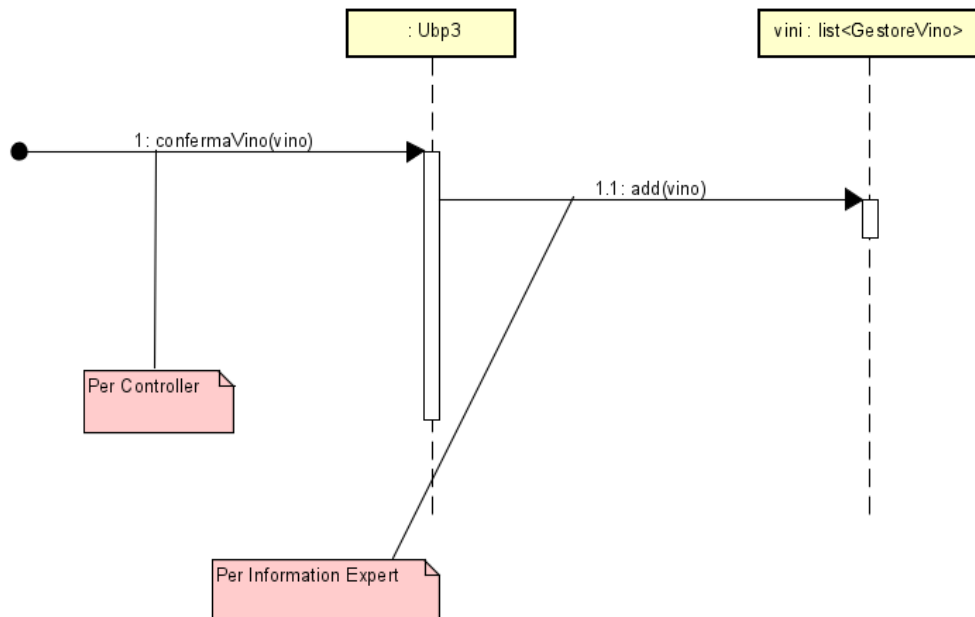
• *inserisciVino*

sd UC1-Interazione 1: inserisciVino



• *confermaVino*

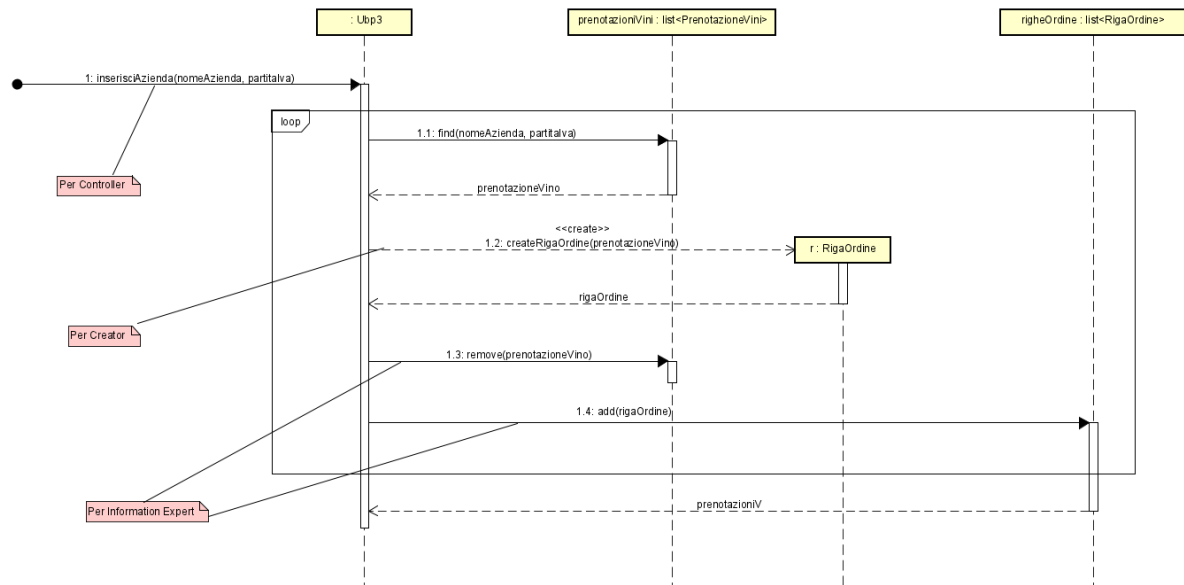
sd UC1-Interazione 2: confermaVino



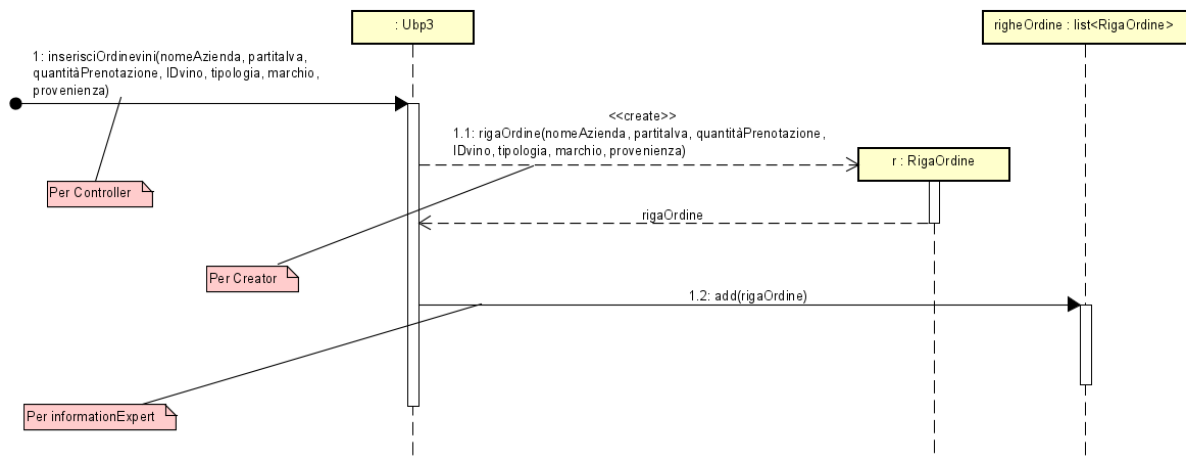
UC2:

• *inserisciAzienda*

sd UC2-Interazione 1: inserisciAzienda

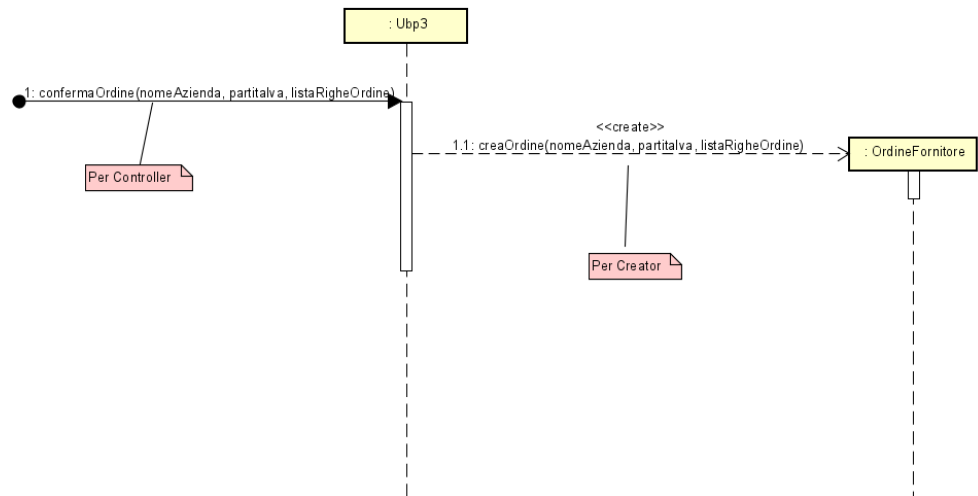


• *inserisciOrdineVini*



- **confermaOrdine**

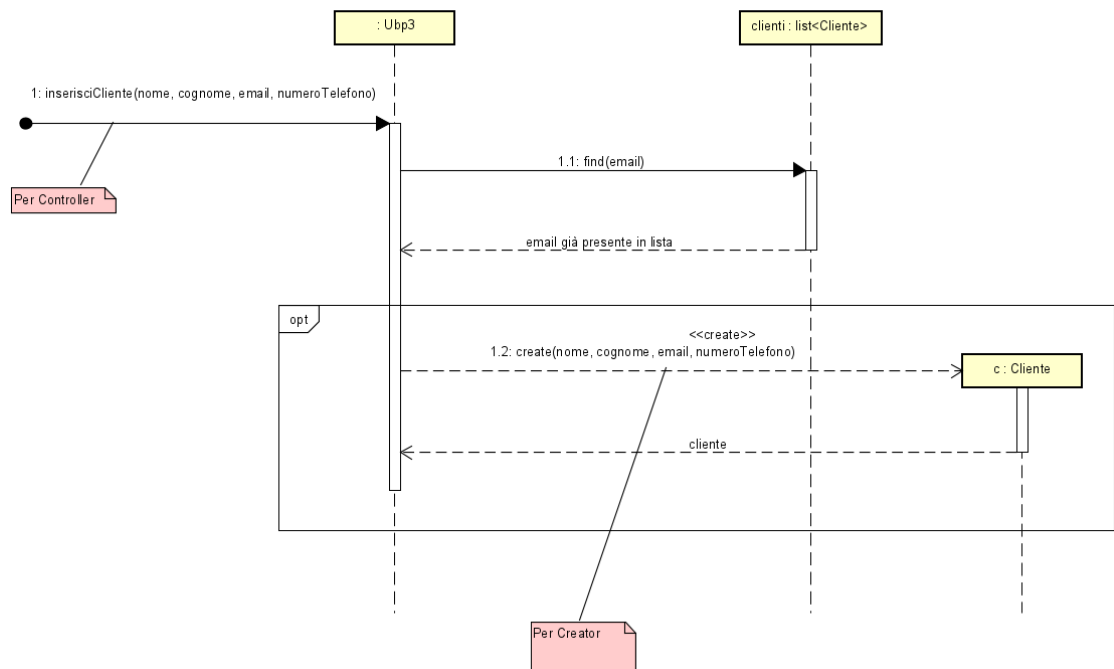
sd UC2-Interazione 3: confermaOrdine



UC3:

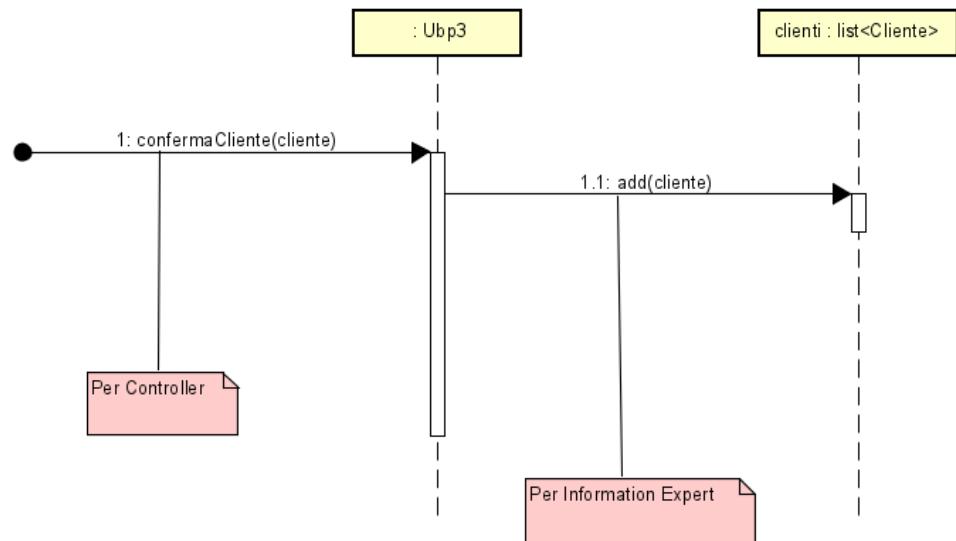
- **inserisciCliente**

sd UC3-Interazione 1: inserisciCliente



- **confermaCliente**

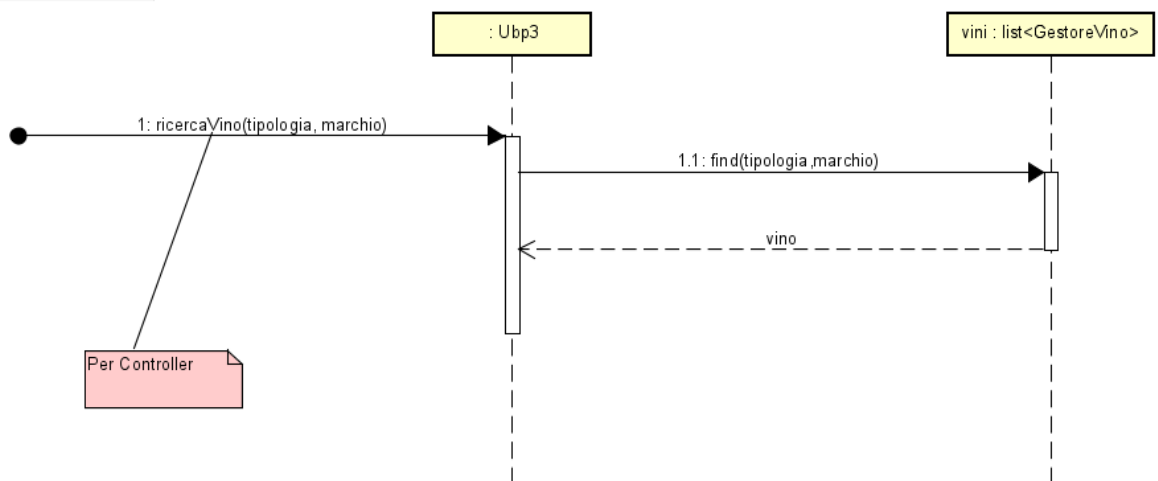
sd UC3-Interazione 2: confermaCliente



UC4:

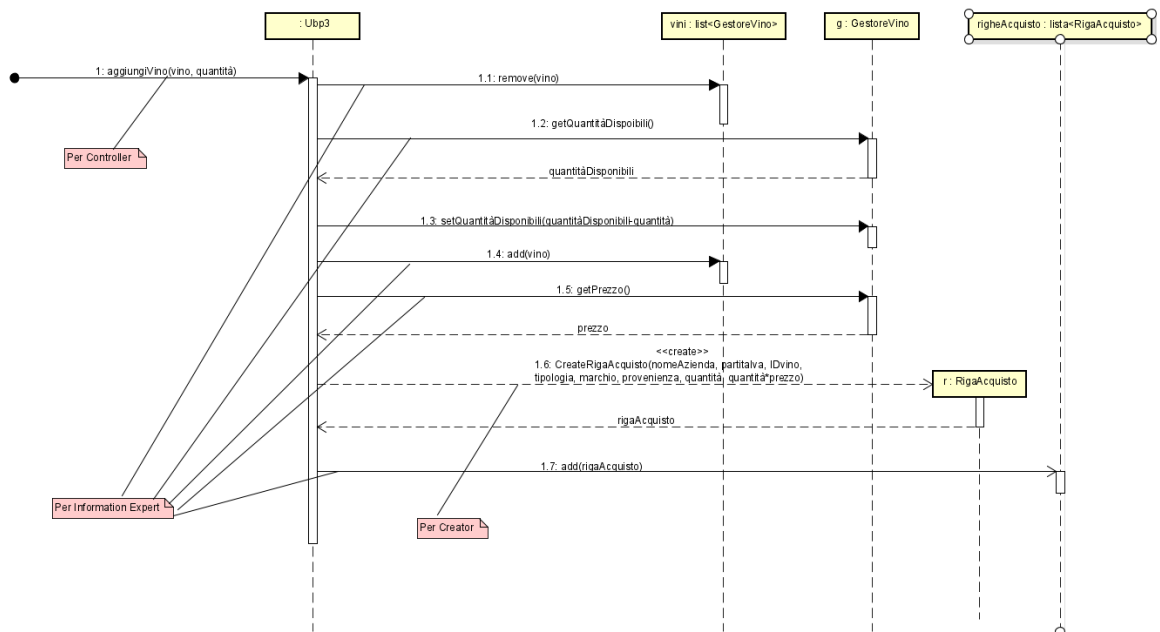
- **ricercaVino**

sd UC4-Interazione 1: ricercaVino



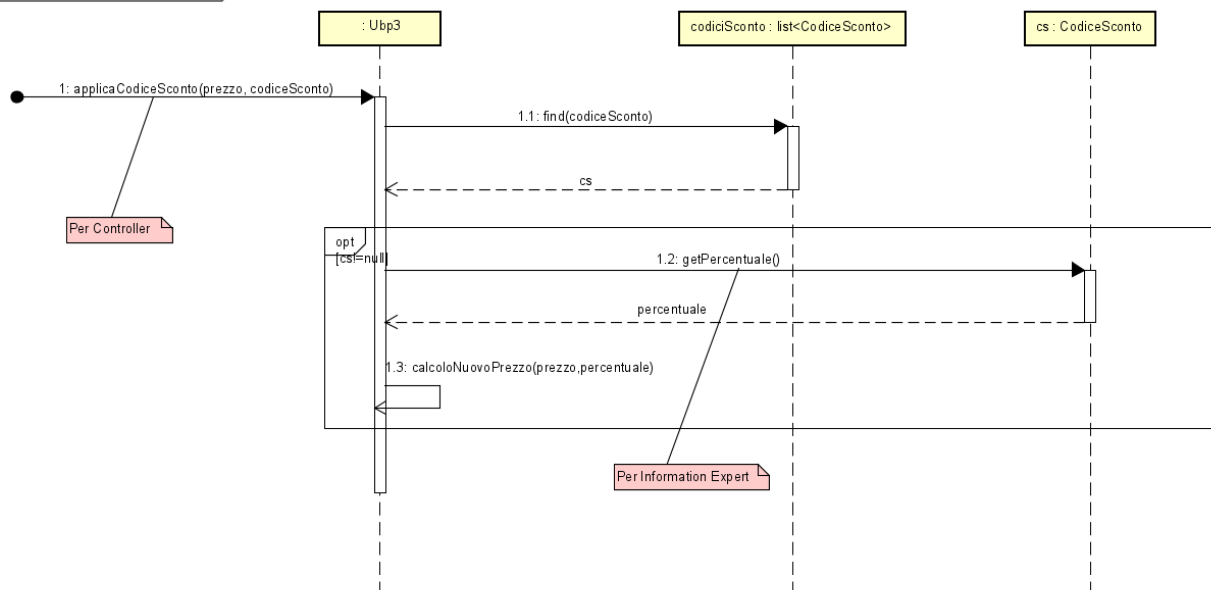
• aggiungiVino

sd UC4-Interazione 2: aggiungiVino



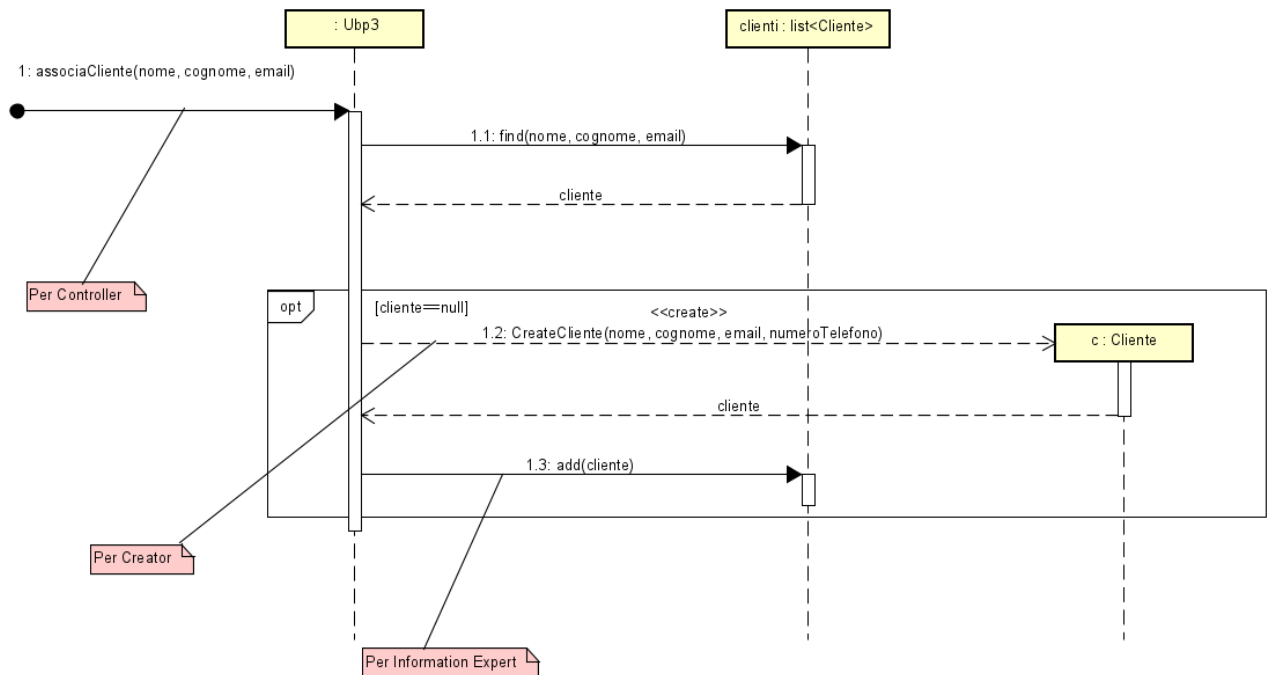
• applicaCodiceSconto

sd UC4-Interazione 3: applicaCodiceSconto



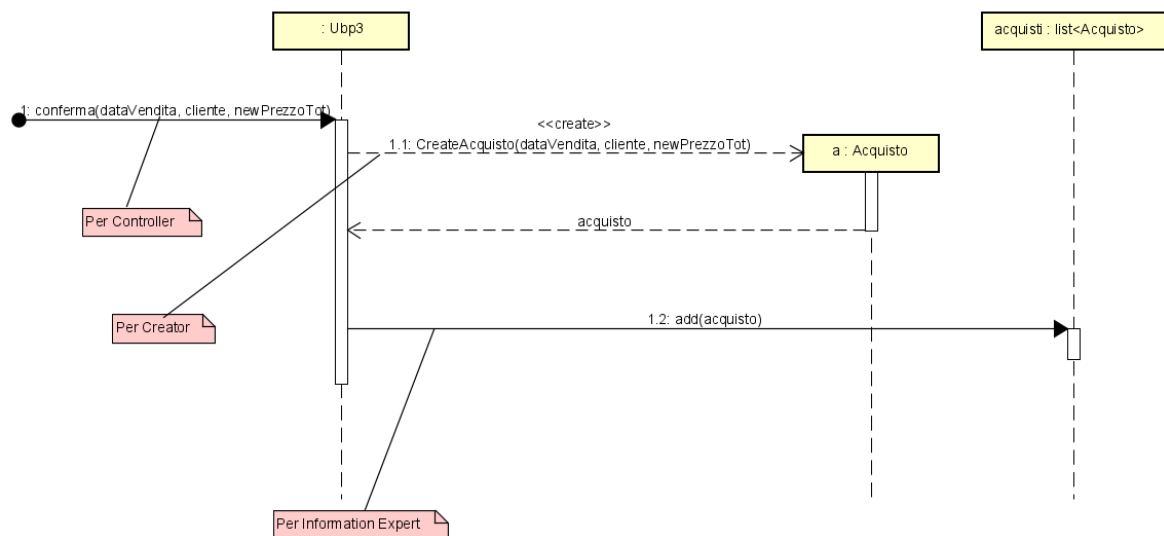
• *associaCliente*

sd UC4-Interazione 4: associaCliente



• *conferma*

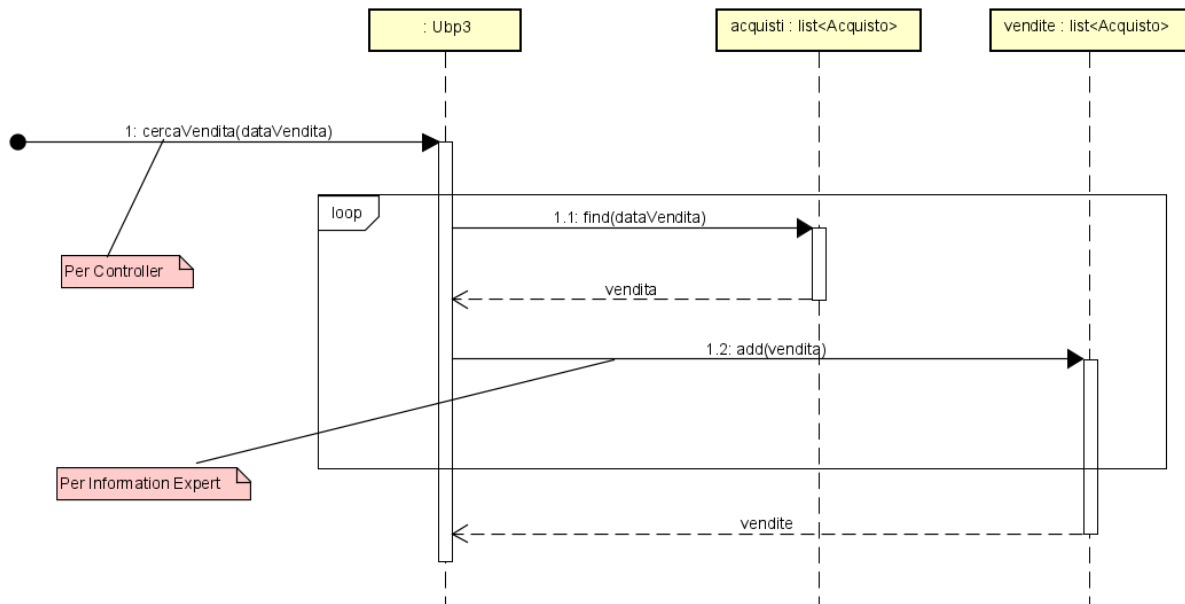
sd UC4-Interazione 5: conferma



UC5:

• *cercaVendita*

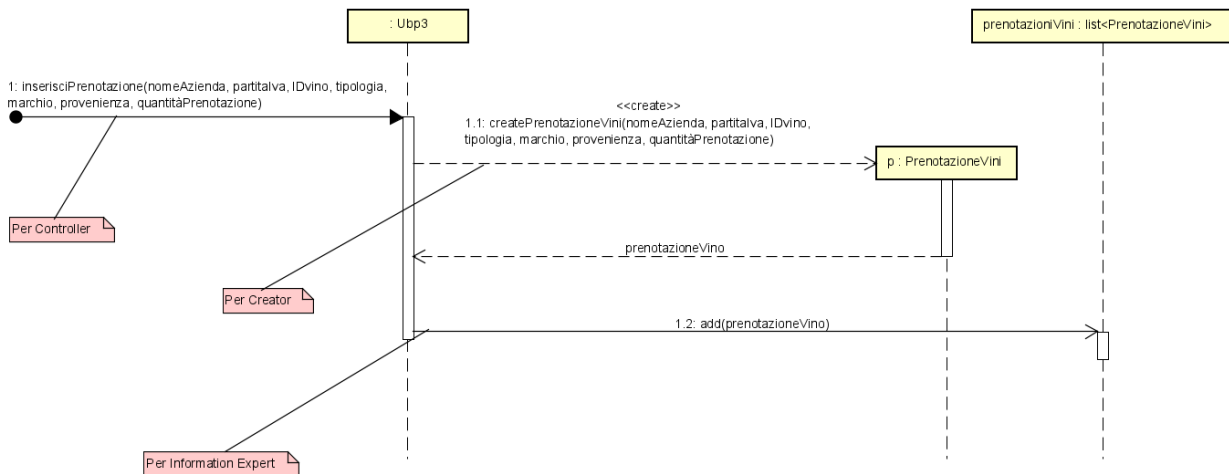
sd UC5-Interazione 1: cercaVendita



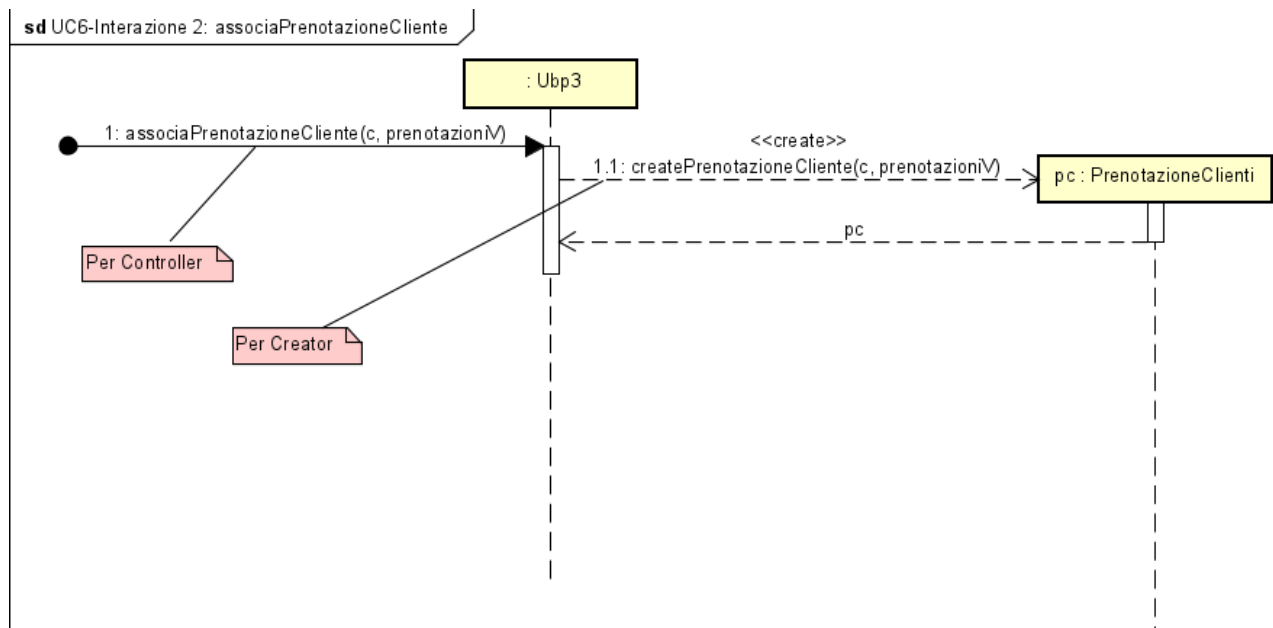
UC6:

• *inserisciPrenotazione*

sd UC6-Interazione 1: inserisciPrenotazione

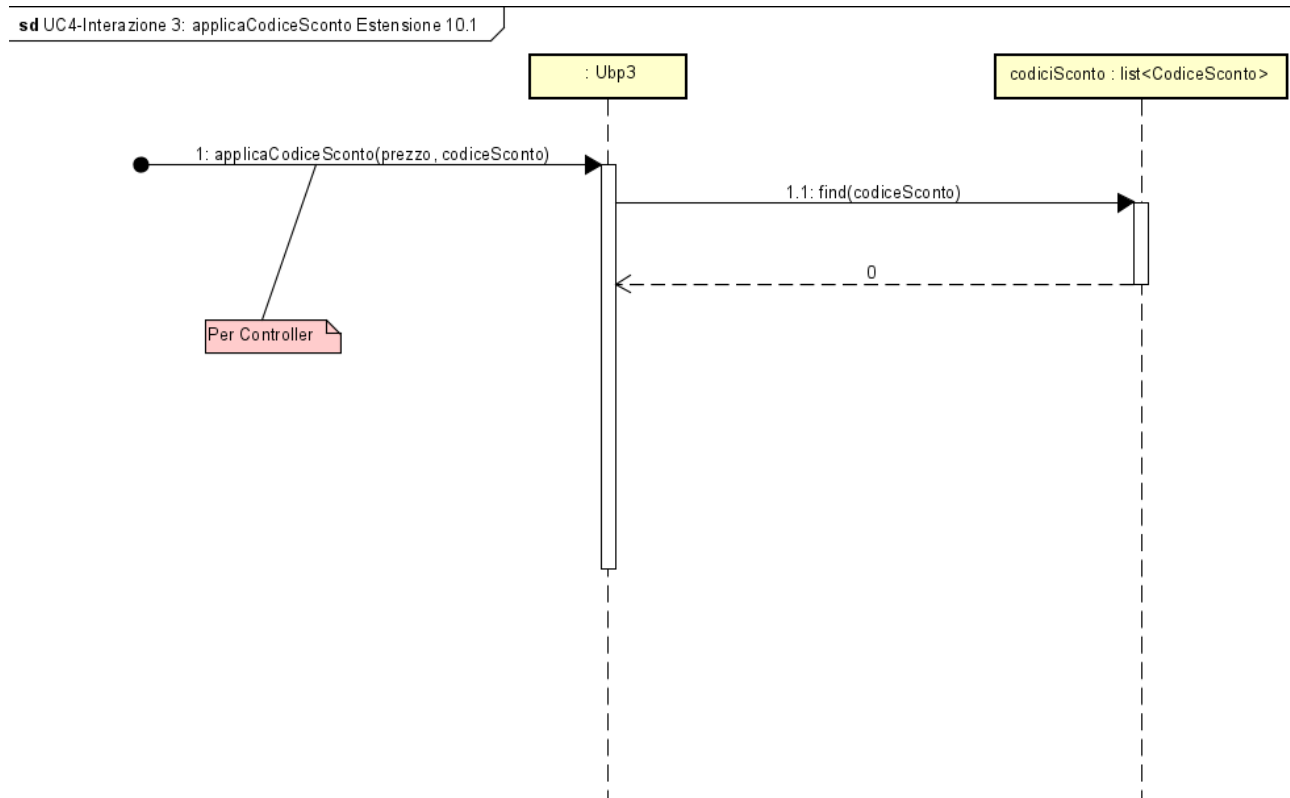


- **associaPrenotazioneCliente**



UC4-Estensione 10.1:

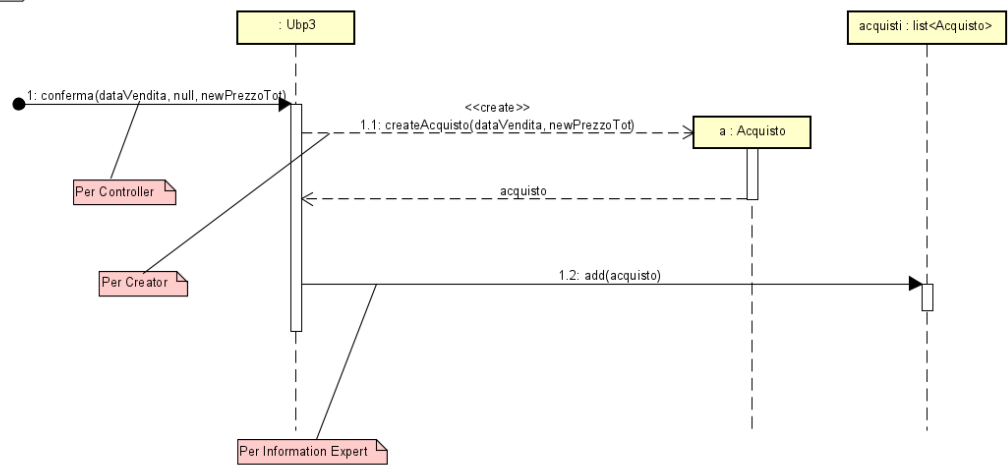
- **applicaCodiceSconto**



UC4-Estensione 11.1:

- **conferma**

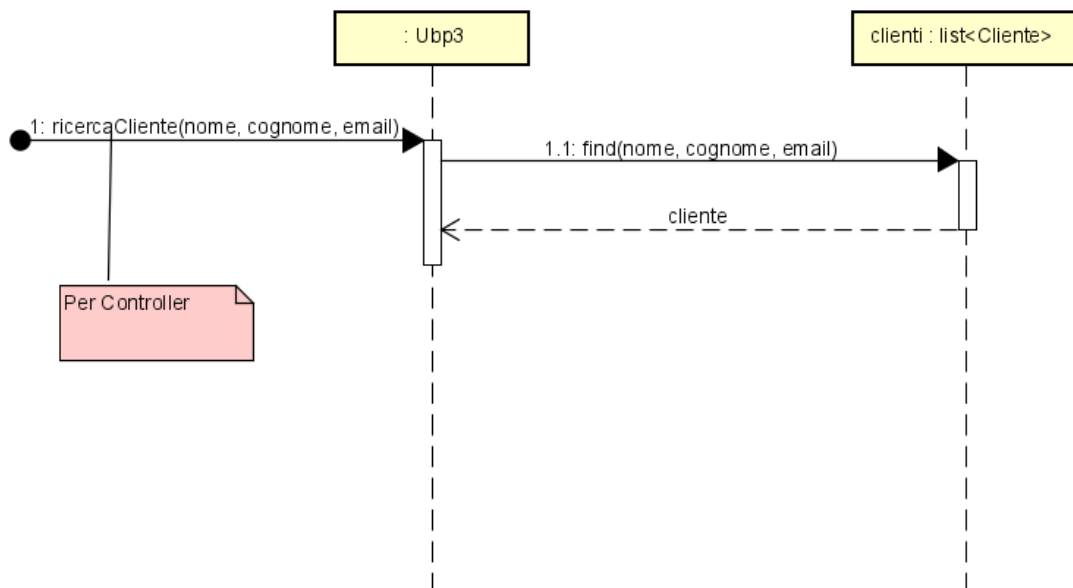
sd UC4-Interazione 4: conferma Estensione 11.1



UC7:

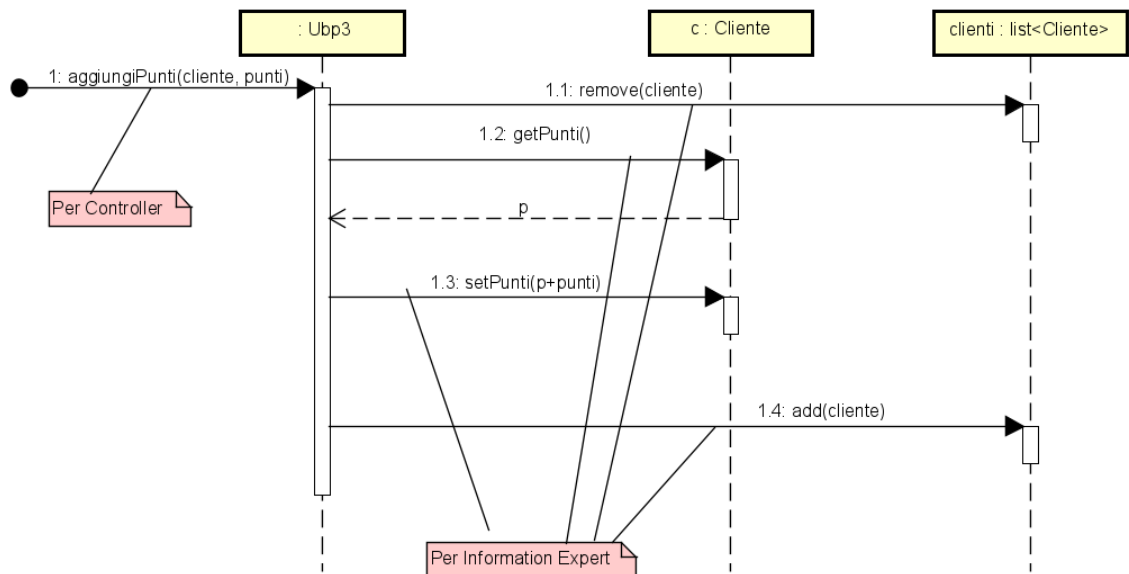
- **ricercaCliente**

sd UC7-Interazione 1: ricercaCliente



- **aggiungiPunti**

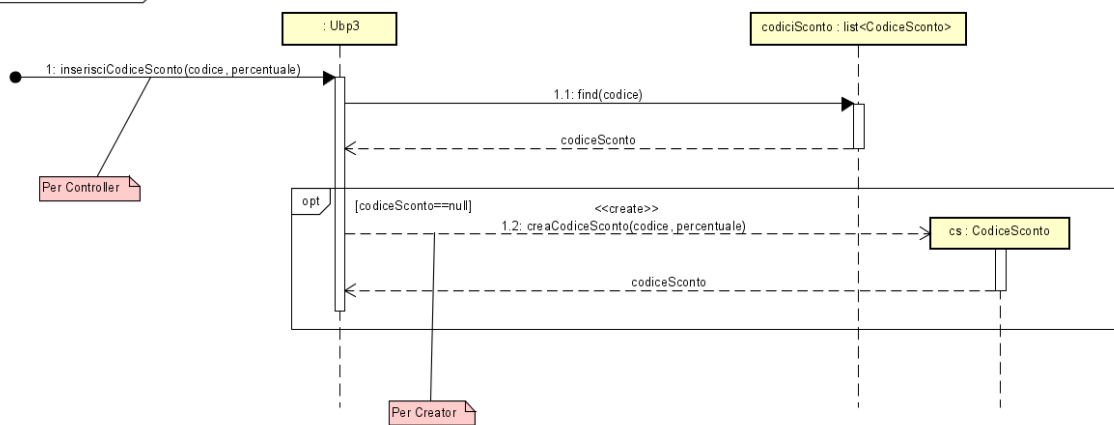
sd UC7-Interazione 2: aggiungiPunti



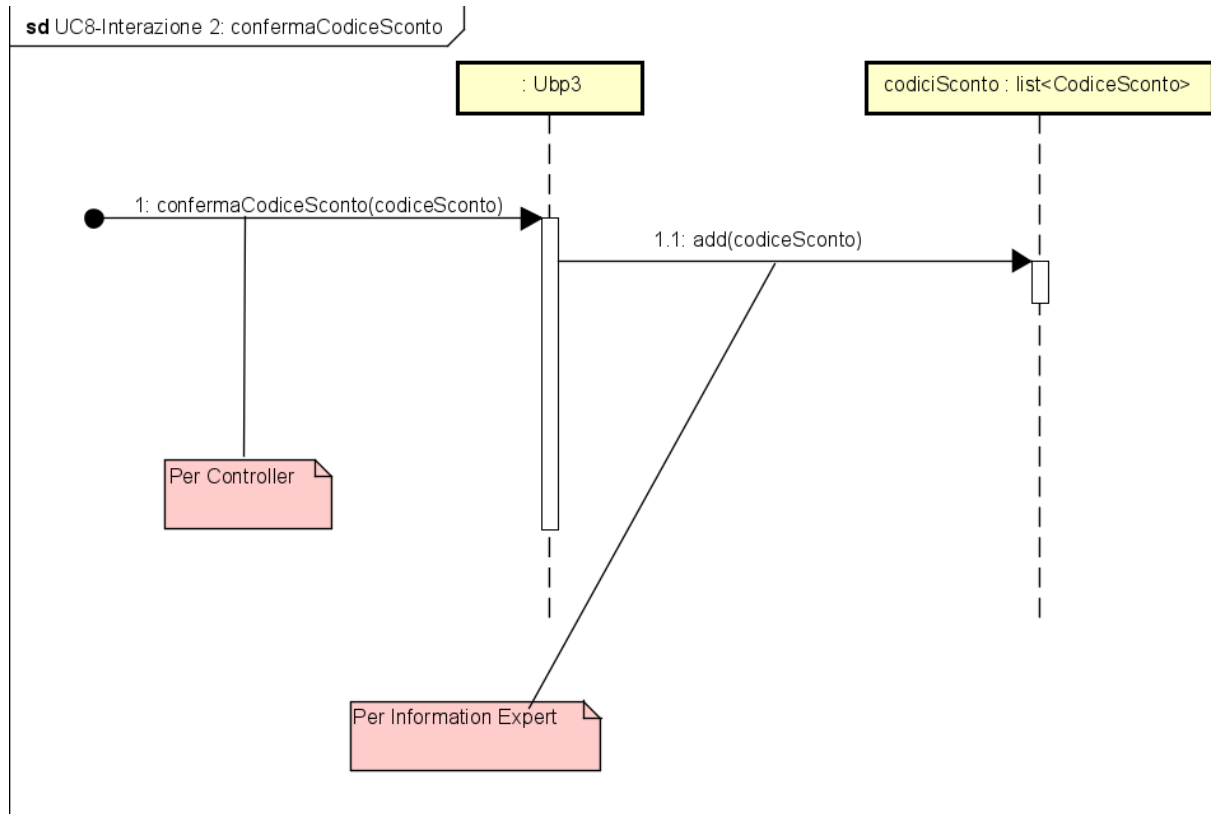
UC8:

- **inserisciCodiceSconto**

sd UC8-Interazione 1: inserisciCodiceSconto

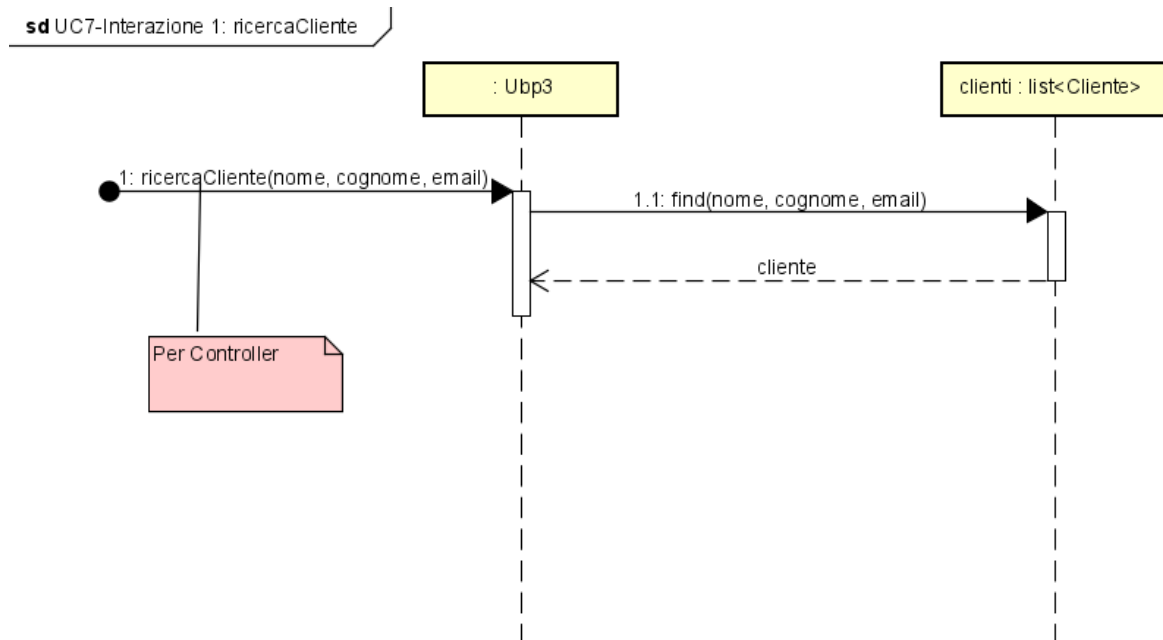


- **confermaCodiceSconto**



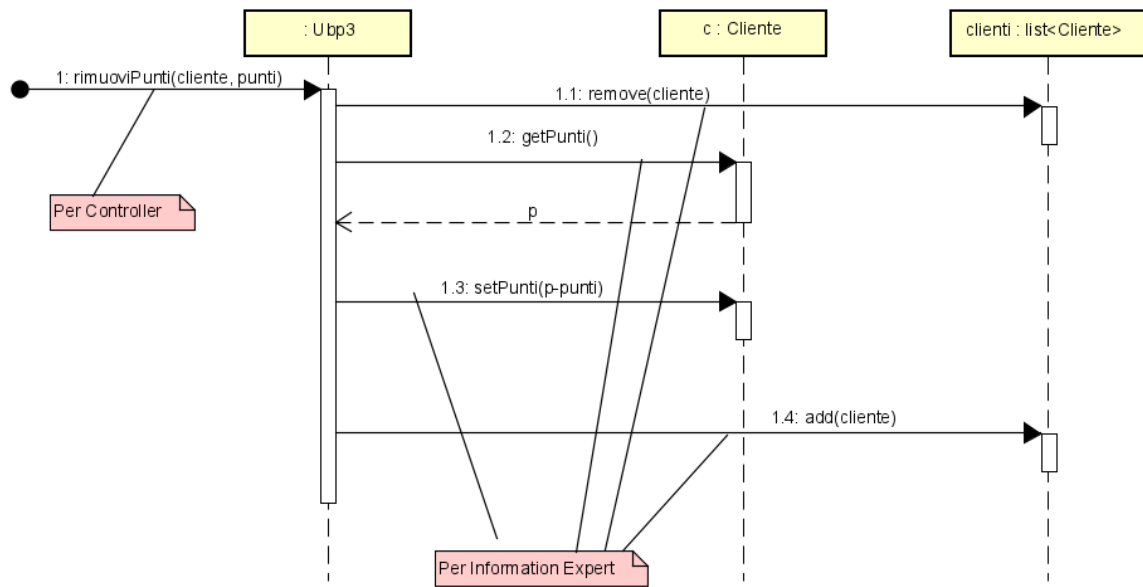
UC9:

- **ricercaCliente**



• *rimuoviPunti*

sd UC9-Interazione 2: rimuoviPunti



4 Testing

4.1 Introduzione

Il **testing** è una fase fondamentale del processo di sviluppo di un programma. Permette di rilevare malfunzionamenti indesiderati al fine di rendere il software realizzato funzionante e funzionale. La probabilità di rilevare gli errori è proporzionale al numero di test eseguiti. E' difficile testare un programma completamente poiché le possibili combinazioni di valori di input da prendere in considerazione sono molto elevati

Esistono due tipologie di test:

- I **test unitari (Unit Test)** → Sono semplici test/prove che vanno a verificare la correttezza direttamente del codice.
- I **test funzionali** → Sono dei test che trattano il sistema come se fosse una scatola nera alla quale danno degli input e verificano la correttezza degli output.

Per eseguire il testing dell'applicazione realizzata, si è scelto di concentrarsi principalmente su test unitari eseguiti tramite l'ausilio di JUnit.

4.2 Introduzione dei casi di test e Testing Unitario

Dopo aver ispezionato il codice sono state scelte le classi e i metodi su cui effettuare il testing. In seguito vengono descritti i test effettuati.

- **Iterazione 1: classe Ubp3Test**
 - **inserisciVino**: vengono passati i parametri richiesti al metodo `inserisciVino` che confronta, tramite ID numerico, se il vino è presente in lista vini. Se il vino è presente restituisce null, altrimenti crea un'istanza `GestoreVino` e la restituisce. Nel test si valuta se l'istanza ritornata è corretta.
 - **confermaVino**: viene creata una nuova istanza `g` di `GestoreVino`, l'istanza viene passata al metodo `confermaVino` che la aggiunge in lista vini. Nel test viene valutato se il nuovo `GestoreVino` è stato correttamente inserito.
 - **inserisciAzienda**: viene passato al metodo `inserisciAzienda` il nome e la partita iva di un'azienda. Il metodo verifica se vi sono dei vini prenotati per l'azienda specificata e in caso positivo li inserisce in una lista e la restituisce. Il test valuta i vini presenti nella lista restituita.

- ***inserisciOrdineVini***: vengono passati al metodo `inserisciOrdineVini` i parametri richiesti. Il metodo crea una nuova istanza `RigaOrdine` e la restituisce. Il test valuta se l'istanza è stata creata correttamente.
- ***inserisciCliente***: vengono passati i parametri richiesti al metodo `inserisciCliente` che verifica se il cliente è presente in lista clienti. In caso positivo restituisce null, altrimenti crea un'istanza `Cliente` e la restituisce. Nel test si valuta se l'istanza ritornata è corretta.
- ***confermaCliente***: viene creata una nuova istanza `cliente` di `Cliente`, questa viene passata al metodo `confermaCliente` che lo inserisce in lista clienti. Nel test viene valutato se il nuovo cliente è stato correttamente inserito in lista.
- **Iterazione 2: classe *Ubp3* 2Test**
 - ***ricercaVino***: viene passato al metodo `ricercaVino`, una tipologia e un marchio. Il metodo ricerca il vino in lista; se è presente lo restituisce altrimenti restituisce null. Il test esamina ciò che è stato restituito dal metodo.
 - ***aggiungiVino***: viene creata una nuova istanza `g` di `GestoreVino` e viene passata con la rispettiva quantità al metodo `aggiungiVino`, che crea un'istanza `RigaAcquisto` e la ritorna. Il test esamina la correttezza dell'istanza restituita.
 - ***associaCliente***: vengono passati al metodo `associaCliente` i dati relativi ad un cliente. Il metodo ricerca il cliente in lista clienti e lo ritorna. Il test valuta se il cliente ritornato è corretto. Per far funzionare correttamente il test, deve essere runnato il test `confermaCliente`, che aggiunge il cliente in lista.
 - ***applicaCodiceSconto***: al metodo `applicaCodiceSconto` vengono passati il prezzo e il codice che si intende applicare; questo metodo valuta se il codice è presente in lista e applica la percentuale di sconto al prezzo. Se il codice è presente ritorna il nuovo prezzo altrimenti ritorna 0. Con il test si confronta il valore ritornato con quello calcolato.
 - ***conferma***: al metodo `conferma` vengono passati: la data dell'acquisto, il cliente ed il prezzo della vendita. Il metodo crea una nuova istanza di `Acquisto` e la inserisce in lista acquisti. Nel test viene valutato se il nuovo acquisto è stato correttamente inserito in lista.

- ***cercaVendita***: viene passato al metodo `cercaVendita` una data. Quest'ultimo ricerca in lista acquisti se sono state effettuate delle vendite in quella data e in caso positivo le aggiunge in una lista e la ritorna. Nel test viene valutato se la lista è stata creata correttamente.
 - ***inserisciPrenotazione***: vengono passati al metodo `inserisciPrenotazione` i parametri richiesti. Il metodo crea un'istanza `PrenotazioneVini`, l'aggiunge in lista e ritorna l'istanza creata. Si valuta nel test se l'istanza è corretta.
 - ***associaPrenotazioneCliente***: viene passato al metodo `associaPrenotazioneCliente` un cliente e una lista `prenotazioniVini`. Il metodo crea una nuova istanza di `PrenotazioneClienti` che viene aggiunta in un file di testo e ritornata. Nel test si valuta se l'istanza ritornata è stata creata correttamente.
- **Iterazione 3: classe Ubp3 3Test**
 - ***ricercaCliente***: al metodo `ricercaCliente` vengono passati i parametri relativi ad un cliente. Il metodo ricerca se il cliente è stato registrato, in caso positivo lo ritorna, altrimenti ritorna null. Nel test si valuta se il cliente ritornato è corretto.
 - ***inserisciCodiceSconto***: vengono passati i parametri richiesti al metodo `inserisciCodiceSconto` che verifica se il codice sconto è presente in lista. In caso positivo restituisce null, altrimenti crea un'istanza `CodiceSconto` e la restituisce. Nel test si valuta se l'istanza ritornata è corretta.
 - ***confermaCodiceSconto***: viene creata un'istanza `CodiceSconto` e viene passata al metodo `confermaCodiceSconto` che la aggiunge alla lista `codiciSconto`. Nel test viene esaminata la lista e viene valutato se il codice sconto è stato aggiunto correttamente.