

Informatizzazione della Libreria “*Amaddeo*”



La libreria *Amaddeo* offre ai propri clienti una vastissima scelta: dai libri scolastici, passando ai best seller, le novità letterarie, libri e manuali per ragazzi, ma anche numerosi testi di piccoli editori e libri di scrittori di nicchia.

1.1 Descrizione del contesto di riferimento

Si vuole realizzare un DBMS per la gestione della libreria "Amaddeo". Il sistema informativo vuole offrire funzionalità sia ai clienti che all'amministratore. La libreria è un esercizio commerciale che attualmente offre ai consumatori standard la vendita di libri. I libri possono essere acquistati sia da clienti persona privati che da clienti associazione; è inoltre disponibile uno sconto su ciascun ordine per coloro che richiedono il rilascio di una tessera.

Il titolare della suddetta libreria ha avanzato la richiesta di creare una base di dati il cui scopo sarà la gestione dei servizi offerti, monitorando i rapporti tra la libreria e i libri presenti in essa, tra la libreria e i suoi fornitori e tra la libreria e la clientela.

Da un'attività di workshop dei requisiti si sono previste per il sistema le seguenti funzionalità:

- un cliente può essere o un privato o un'associazione;
- la vendita di un libro può essere fatta a qualsiasi cliente lo richieda; per i libri scontati, la vendita ai clienti persona o associazione può avvenire solo ai clienti muniti di una tessera personale che certifichi il/i precedente/i acquisto/i;
- i privati o le associazioni si possono pertanto dividere in tesserati e non tesserati; i clienti non tesserati (o clienti occasionali) possono solo acquistare un libro a prezzo pieno; i clienti tesserati possono invece ricevere uno sconto;
- la vendita di un libro a prezzo scontato può quindi essere concessa solo ai clienti tesserati;
- i clienti persona che vorranno avere una tessera dovranno fornire i dati personali ed esporre un documento di riconoscimento; le informazioni da memorizzare per ogni tessera saranno quindi: *Nome, Cognome, Data di nascita, Indirizzo, Telefono, Numero del documento di riconoscimento*;

- per i clienti associazione che vorranno una tessera, si prevedono invece i seguenti attributi: *Denominazione, Indirizzo, Telefono*;
- il sistema deve mantenere traccia dei clienti che acquistano un libro;
- per ciascuna vendita è necessario emettere un documento fiscale;
- per ogni libro sarà importante memorizzare le seguenti informazioni: *Codice Libro, Titolo, Casa Editrice, Autore, Genere, Anno, Quantità Iniziale, Quantità Corrente, Prezzo*;
- “*Quantità Iniziale*” è il numero iniziale di copie disponibili per un libro;
- “*Quantità Corrente*” è il numero corrente di copie disponibili per un libro, inizialmente uguale a “*Quantità Iniziale*”;
- di ogni libro possono esistere più copie; tale numero di copie deve essere memorizzato nell’attributo “*Quantità Iniziale*”;
- inizialmente l’attributo “*Quantità Corrente*” è uguale a “*Quantità Iniziale*”, ma a differenza di quest’ultimo è destinato a variare nel tempo;
- l’approvvigionamento da parte della libreria dei libri avviene tramite i relativi fornitori;
- i rapporti che possono instaurarsi tra libreria e fornitori sono i seguenti:
 1. la libreria può ordinare più copie di un libro ad un fornitore;
 2. la libreria può ritirare da un fornitore le copie di un libro precedentemente ordinato; il numero di copie ordinato e ritirato è detto “*Quantità Iniziale*” del libro;
 3. la libreria può restituire tutte le copie di un libro che non è più vendibile in quanto sostituito da una nuova edizione;
 4. la libreria può restituire le copie di un libro rimaste invendute;
 5. per ogni fornitore è importante memorizzare le seguenti informazioni: *Codice Fornitore, Partita IVA, Nome, Indirizzo, Telefono*;

- le vendite dei libri avvengono all'interno della libreria durante il seguente orario di servizio:

Lunedì	09:00 – 13:00	16:00 – 20:00
Martedì	09:00 – 13:00	16:00 – 20:00
Mercoledì	09:00 – 13:00	16:00 – 20:00
Giovedì	09:00 – 13:00	16:00 – 20:00
Venerdì	09:00 – 13:00	16:00 – 20:00
Sabato	09:00 – 13:00	Chiuso
Domenica	Chiuso	Chiuso

- i costi per la vendita variano articolo per articolo secondo un prezzo di listino;
- una tessera può essere creata solo dall'amministratore.

1.2 Analisi dei requisiti

Un requisito è una specifica che deve essere implementata nel sistema. Per quanto riguarda il sistema informativo a supporto della libreria presa in considerazione in questo progetto, si possono individuare i seguenti **requisiti funzionali** (espressi secondo la formulazione “<id> il < sistema > dovrà <funzione>”):

- **RF1:** il sistema dovrà gestire le attività “CRUD” sui fornitori (casa editrice);
- **RF2:** il sistema dovrà gestire le attività “CRUD” sull'ordine di un libro ad un fornitore;
- **RF3:** il sistema dovrà gestire le attività “CRUD” sul ritiro dell'ordine di un libro da un fornitore;
- **RF4:** il sistema dovrà gestire le attività “CRUD” sulla restituzione di un libro ad un fornitore;
- **RF5:** il sistema dovrà gestire le attività “CRUD” su un libro di qualsiasi genere;

- **RF6:** il sistema dovrà gestire le attività “CRUD” su un cliente, sia esso persona fisica o associazione (che acquista libri per i propri utenti);
- **RF7:** il sistema dovrà gestire le attività “CRUD” su una tessera (sconti);
- **RF8:** il sistema dovrà gestire le attività “CRUD” sulla vendita di un libro;
- **RF9:** il sistema dovrà gestire le attività “CRUD” su un documento fiscale;
- **RF10:** il sistema dovrà aggiornare la quantità corrente disponibile per la vendita dopo ciascuna vendita;
- **RF11:** il sistema dovrà gestire un meccanismo di autenticazione di un cliente tesserato;
- **RF12:** il sistema dovrà gestire un meccanismo di autenticazione dell’amministratore;
- **RF13:** il sistema dovrà consentire ad un cliente tesserato soltanto la gestione dei propri dati;
- **RF14:** il sistema dovrà consentire all’amministratore l’accesso, sia in lettura che in scrittura, a tutti i dati memorizzati nel corrispondente database.

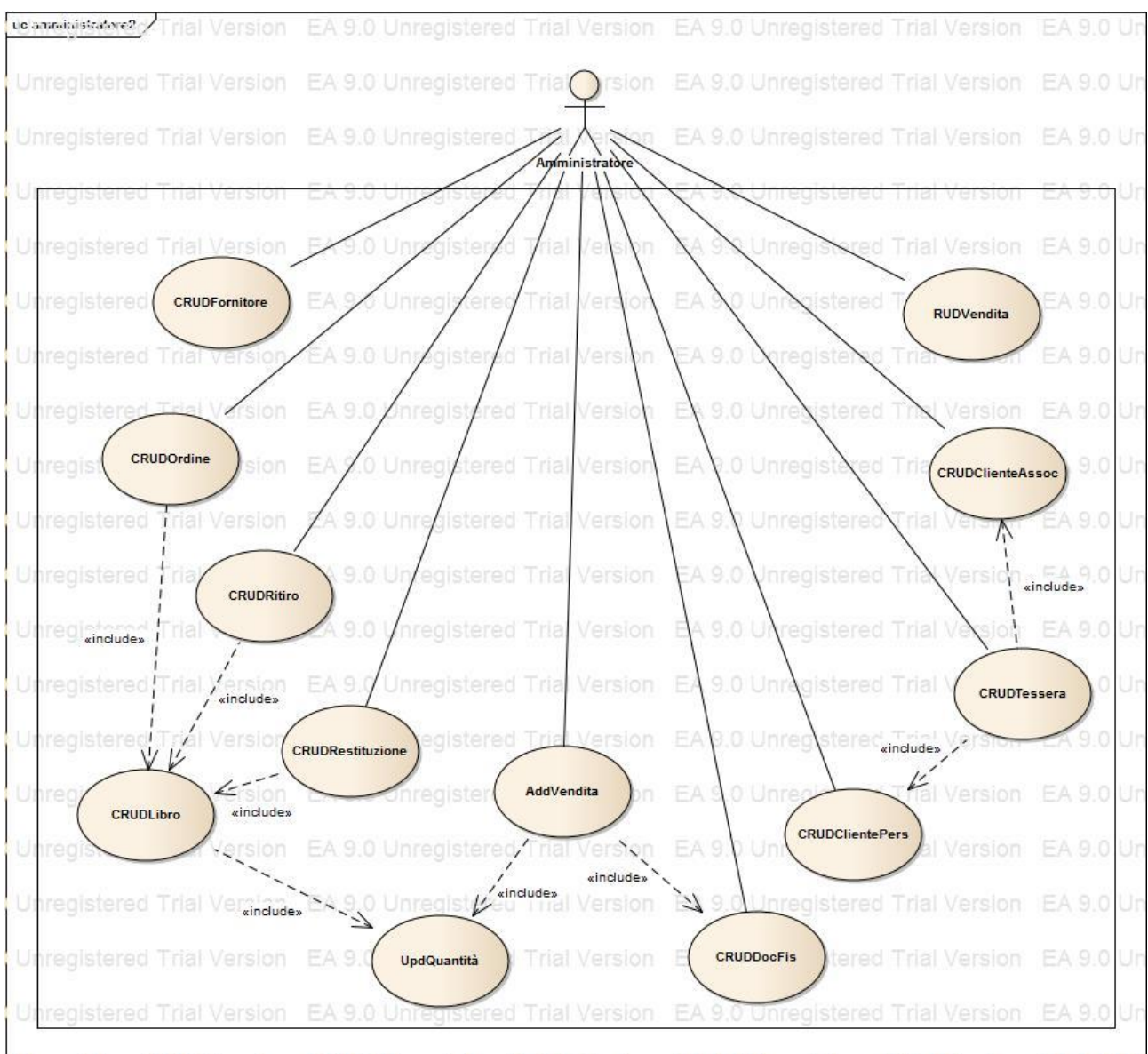
Si possono inoltre individuare i seguenti **requisiti non funzionali** (vincoli di implementazione):

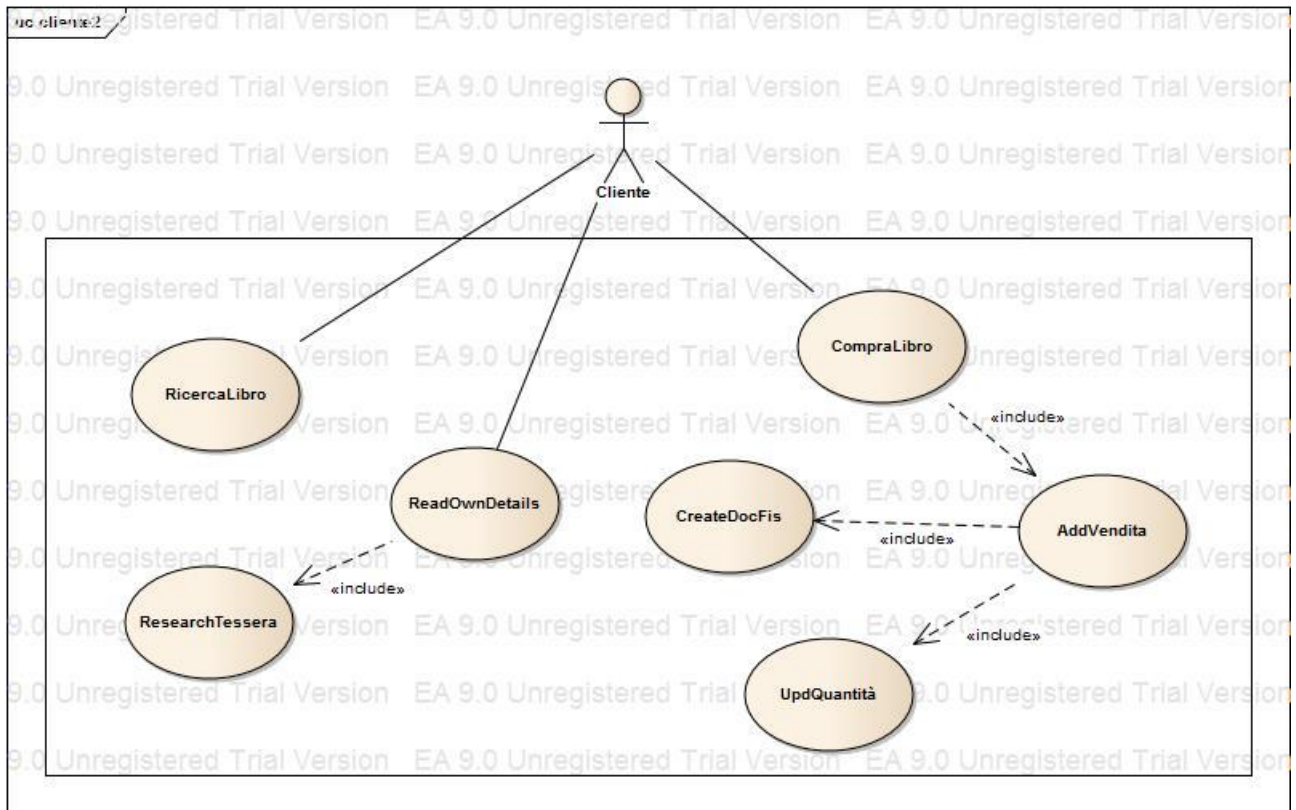
- **RNF1:** il sistema dovrà possedere una password di almeno otto caratteri di cui almeno due alfanumerici per i clienti tesserati e l’amministratore;
- **RNF2:** il sistema dovrà memorizzare tutti i dati personali necessari per ogni tessera richiesta;
- **RNF3:** il sistema dovrà impedire lo sconto su un libro ad un cliente persona/associazione che non risulti tesserato/a;
- **RNF4:** il sistema dovrà consentire lo sconto su un libro solo ad un cliente persona/associazione tesserato/a;
- **RNF5:** il sistema dovrà impedire la vendita di un libro vendibile la cui quantità corrente sia zero;

- **RNF6:** il sistema dovrà consentire la creazione di una tessera solo all'amministratore.

1.3 Diagramma dei casi d'uso

Sulla base dei requisiti funzionali espressi precedentemente si è ora in grado di costruire il diagramma dei casi d'uso per il sistema in questione. Gli attori previsti sono due: **cliente** ed **amministratore**.





Come si evince dalle figure sono presenti i seguenti 14 casi d'uso istanziati:

- | | |
|----------------------------------|----------------------------------|
| 1. CU1: CRUDFornitore; | 8. CU8: CRUDLibro; |
| 2. CU2: CRUDOrdine; | 9. CU9: RUDVendita; |
| 3. CU3: CRUDRitiro; | 10. CU10: AddVendita; |
| 4. CU4: CRUDRestituzione; | 11. CU11: CRUDDocFis; |
| 5. CU5: CRUDClienteAssoc; | 12. CU12: RicercaLibro; |
| 6. CU6: CRUDClientePers; | 13. CU13: CompraLibro; |
| 7. CU7: CRUDTessera; | 14. CU14: ReadOwnDetails. |

Si fa notare che per l'attore **amministratore** abbiamo il seguente caso d'uso non istanziato: UpdQuantità, in quanto l'aggiornamento della quantità corrente può essere istanziato direttamente dal caso d'uso CRUDLibro.

Caso d'uso: CRUDFornitore
ID: CU1
Breve descrizione: Questo caso d'uso consente l'inserimento, la rimozione, la modifica e la ricerca di informazioni relative ad un fornitore.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD sui dati relativi ai fornitori; 2. <i>if</i> l'attore primario vuole inserire un nuovo fornitore 2.1. l'attore primario inserisce i dati relativi al nuovo fornitore; 2.2. il sistema memorizza tali dati nel database; 3. <i>else if</i> l'attore primario vuole ricercare nel database i dati di un fornitore 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca il fornitore specificato nel database; 3.3. <i>if</i> il fornitore specificato è presente nel database 3.3.1. il sistema visualizza i corrispettivi dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole aggiornare i dati di un fornitore 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca il fornitore specificato nel database; 4.3. <i>if</i> il fornitore specificato è presente nel database 4.3.1. l'attore primario specifica i nuovi dati da inserire; 4.3.2. il sistema procede con l'aggiornamento dei dati; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore; 5. <i>else if</i> l'attore primario vuole rimuovere i dati di un fornitore 5.1. l'attore primario specifica una chiave di ricerca; 5.2. il sistema ricerca il fornitore specificato nel database; 5.3. <i>if</i> il fornitore specificato è presente nel database 5.3.1. il sistema procede con la sua rimozione; 5.4. <i>else</i> 5.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: CRUDOrdine
ID: CU2
Breve descrizione: Questo caso d'uso consente l'inserimento, la rimozione, la modifica e la ricerca di informazioni relative ad un ordine di un libro ad un fornitore.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD su un ordine di un libro ad un fornitore; 2. <i>if</i> l'attore primario vuole inserire un nuovo ordine di un libro ad un fornitore 2.1. l'attore primario inserisce i dati relativi al nuovo ordine di un libro; 2.2. il sistema memorizza tali dati nel database; 2.3. il libro dell'ordine è memorizzato con un suo codice nel database con quantità iniziale pari alla quantità ordinata e quantità corrente uguale a 0 [Include(CRUDLibro)]; 3. <i>else if</i> l'attore primario vuole ricercare nel database i dati di un ordine di un libro 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca l'ordine specificato nel database; 3.3. <i>if</i> l'ordine specificato è presente nel database 3.3.1. il sistema visualizza i corrispettivi dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole aggiornare i dati di ordine di un libro 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca l'ordine specificato nel database; 4.3. <i>if</i> l'ordine specificato è presente nel database 4.3.1. l'attore primario specifica i nuovi dati da inserire; 4.3.2. il sistema procede con l'aggiornamento dei dati [Include(CRUDLibro)]; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore; 5. <i>else if</i> l'attore primario vuole rimuovere i dati di un ordine di un libro 5.1. l'attore primario specifica una chiave di ricerca; 5.2. il sistema ricerca l'ordine specificato nel database; 5.3. <i>if</i> l'ordine specificato è presente nel database 5.3.1. il sistema procede con la sua rimozione; 5.3.2. il libro dell'ordine rimosso è rimosso dal database [Include(CRUDLibro)]; 5.4. <i>else</i> 5.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Nel caso di inserimento, aggiornamento e rimozione di un ordine di un libro, l'archivio dei libri è stato aggiornato tramite CRUDLibro.
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: CRUDRitiro
ID: CU3
Breve descrizione: Questo caso d'uso consente l'inserimento, la rimozione, la modifica e la ricerca di informazioni relative ad un ritiro di un ordine di un libro.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD relativa ad un ritiro di un ordine di un libro; 2. <i>if</i> l'attore primario vuole inserire un nuovo ritiro di un libro da un fornitore 2.1. l'attore primario inserisce i dati relativi al nuovo ritiro; 2.2. il sistema memorizza tali dati nel database; 2.3. la quantità corrente del libro dell'ordine ritirato è portata da 0 alla quantità iniziale [Include(CRUDLibro)]; 3. <i>else if</i> l'attore primario vuole ricercare nel database i dati di un ritiro di un libro 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca il ritiro specificato nel database; 3.3. <i>if</i> il ritiro specificato è presente nel database 3.3.1. il sistema visualizza i corrispettivi dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole aggiornare i dati di un ritiro di un libro 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca il ritiro specificato nel database; 4.3. <i>if</i> il ritiro specificato è presente nel database 4.3.1. l'attore primario specifica i nuovi dati da inserire; 4.3.2. il sistema procede con l'aggiornamento dei dati [Include(CRUDLibro)]; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore; 5. <i>else if</i> l'attore primario vuole rimuovere i dati di un ritiro di un libro 5.1. l'attore primario specifica una chiave di ricerca; 5.2. il sistema ricerca il ritiro specificato nel database; 5.3. <i>if</i> il ritiro specificato è presente nel database 5.3.1. il sistema procede con la sua rimozione; [Include(CRUDLibro)]; 5.4. <i>else</i> 5.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Nel caso di inserimento, aggiornamento e rimozione di un ritiro di un libro, l'archivio dei libri è stato aggiornato tramite CRUDLibro
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: CRUDRestituzione
ID: CU4
Breve descrizione: Questo caso d'uso consente l'inserimento, la rimozione, la modifica e la ricerca di informazioni relative ad una restituzione di un libro ad un fornitore.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD sulla restituzione di un libro ad un fornitore; 2. <i>if</i> l'attore primario vuole inserire una nuova restituzione di un libro ad un fornitore 2.1. l'attore primario inserisce i dati relativi alla nuova restituzione; 2.2. il sistema memorizza tali dati nel database; 2.3. la quantità corrente del libro restituito è portata a 0 [<i>Include</i> (CRUDLibro)]; 3. <i>else if</i> l'attore primario vuole ricercare nel database i dati di una restituzione di un libro ad un fornitore 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca la restituzione specificata nel database; 3.3. <i>if</i> la restituzione specificata è presente nel database 3.3.1. il sistema visualizza i corrispettivi dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole aggiornare i dati di una restituzione di un libro ad un fornitore 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca la restituzione specificata nel database; 4.3. <i>if</i> la restituzione specificata è presente nel database 4.3.1. l'attore primario specifica i nuovi dati da inserire; 4.3.2. il sistema procede con l'aggiornamento dei dati [<i>Include</i> (CRUDLibro)]; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore; 5. <i>else if</i> l'attore primario vuole rimuovere i dati di una restituzione di un libro ad un fornitore 5.1. l'attore primario specifica una chiave di ricerca; 5.2. il sistema ricerca la restituzione specificata nel database; 5.3. <i>if</i> la restituzione specificata è presente nel database 5.3.1. il sistema procede con la sua rimozione [<i>Include</i> (CRUDLibro)]; 5.4. <i>else</i> 5.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Nel caso di inserimento, aggiornamento e rimozione di una restituzione di un libro, l'archivio dei libri è stato aggiornato tramite CRUDLibro
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: CRUDClienteAssoc
ID: CU5
Breve descrizione: Questo caso d'uso consente l'inserimento, la rimozione, la modifica e la ricerca di informazioni relative ad una associazione culturale.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD sui dati relativi ad un cliente associazione; 2. <i>if</i> l'attore primario vuole inserire un nuovo cliente associazione 2.1. l'attore primario inserisce i dati relativi al nuovo cliente associazione; 2.2. il sistema memorizza tali dati nel database; 3. <i>else if</i> l'attore primario vuole ricercare nel database i dati di un cliente associazione 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca il cliente associazione specificato nel database; 3.3. <i>if</i> il cliente associazione specificato è presente nel database 3.3.1. il sistema visualizza i corrispettivi dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole aggiornare i dati di un cliente associazione 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca il cliente associazione specificato nel database; 4.3. <i>if</i> il cliente associazione specificato è presente nel database 4.3.1. l'attore primario specifica i nuovi dati da inserire; 4.3.2. il sistema procede con l'aggiornamento dei dati; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore; 5. <i>else if</i> l'attore primario vuole rimuovere i dati di un cliente associazione 5.1. l'attore primario specifica una chiave di ricerca; 5.2. il sistema ricerca il cliente associazione specificato nel database; 5.3. <i>if</i> il cliente associazione specificato è presente nel database 5.3.1. il sistema procede con la sua rimozione; 5.4. <i>else</i> 5.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: CRUDClientePers
ID: CU6
Breve descrizione: Questo caso d'uso consente l'inserimento, la rimozione, la modifica e la ricerca di informazioni relative ad un cliente persona.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD sui dati relativi ad un cliente persona; 2. <i>if</i> l'attore primario vuole inserire un nuovo cliente persona 2.1. l'attore primario inserisce i dati relativi al nuovo cliente persona; 2.2. il sistema memorizza tali dati nel database; 3. <i>else if</i> l'attore primario vuole ricercare nel database i dati di un cliente persona 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca il cliente persona specificato nel database; 3.3. <i>if</i> il cliente persona specificato è presente nel database 3.3.1. il sistema visualizza i corrispettivi dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole aggiornare i dati di un cliente persona 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca il cliente persona specificato nel database; 4.3. <i>if</i> il cliente persona specificato è presente nel database 4.3.1. l'attore primario specifica i nuovi dati da inserire; 4.3.2. il sistema procede con l'aggiornamento dei dati; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore; 5. <i>else if</i> l'attore primario vuole rimuovere i dati di un cliente persona 5.1. l'attore primario specifica una chiave di ricerca; 5.2. il sistema ricerca il cliente persona specificato nel database; 5.3. <i>if</i> il cliente persona specificato è presente nel database 5.3.1. il sistema procede con la sua rimozione; 5.4. <i>else</i> 5.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: CRUdTessera
ID: CU7
Breve descrizione: Questo caso d'uso consente l'inserimento, la rimozione, la modifica e la ricerca di informazioni relative ad una tessera di un cliente persona o associazione.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD sui dati relativi ad una tessera di un cliente persona o associazione; 2. <i>if</i> l'attore primario vuole inserire una nuova tessera 2.1. <i>if</i> il cliente persona o associazione è già tracciato nel database 2.1.1. l'attore primario crea subito la tessera 2.2. <i>else</i> 2.2.1. l'attore primario inserisce i dati relativi alla nuova tessera; 2.3. il sistema memorizza tali dati nel database; 3. <i>else if</i> l'attore primario vuole ricercare nel database i dati di una tessera 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca la tessera specificata nel database; 3.3. <i>if</i> la tessera specificata è presente nel database 3.3.1. il sistema visualizza i corrispettivi dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole aggiornare i dati di una tessera 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca la tessera specificata nel database; 4.3. <i>if</i> la tessera specificata è presente nel database 4.3.1. l'attore primario specifica i nuovi dati da inserire; 4.3.2. il sistema procede con l'aggiornamento dei dati; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore; 5. <i>else if</i> l'attore primario vuole rimuovere i dati di una tessera 5.1. l'attore primario specifica una chiave di ricerca; 5.2. il sistema ricerca la tessera specificata nel database; 5.3. <i>if</i> la tessera specificata è presente nel database 5.3.1. il sistema procede con la sua rimozione; 5.4. <i>else</i> 5.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Se le modifiche ad una tessera sono anche modifiche dei dati del cliente persona o associazione, i dati di quel cliente persona o associazione sono stati aggiornati
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: CRUDLibro
ID: CU8
Breve descrizione: Questo caso d'uso consente l'inserimento, la rimozione, la modifica e la ricerca di informazioni relative ad un libro.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD sui dati relativi ad un libro; 2. <i>if</i> l'attore primario vuole inserire un nuovo libro 2.1. l'attore primario inserisce i dati relativi al nuovo libro; 2.2. il sistema memorizza tali dati nel database; 3. <i>else if</i> l'attore primario vuole ricercare nel database i dati di un libro 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca il libro specificato nel database; 3.3. <i>if</i> il libro specificato è presente nel database 3.3.1. il sistema visualizza i corrispettivi dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole aggiornare i dati di un libro 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca il libro specificato nel database; 4.3. <i>if</i> il libro specificato è presente nel database 4.3.1. l'attore primario specifica i nuovi dati da inserire; 4.3.2. il sistema procede con l'aggiornamento dei dati; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore; 5. <i>else if</i> l'attore primario vuole rimuovere i dati di un libro 5.1. l'attore primario specifica una chiave di ricerca; 5.2. il sistema ricerca il libro specificato nel database; 5.3. <i>if</i> il libro specificato è presente nel database 5.3.1. il sistema procede con la sua rimozione; 5.4. <i>else</i> 5.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: RUDVendita
ID: CU9
Breve descrizione: Questo caso d'uso consente la rimozione, la modifica e la ricerca di informazioni relative ad una vendita.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione RUD sui dati relativi ad una vendita; 2. <i>if</i> l'attore primario vuole ricercare nel database i dati di una vendita 2.1. l'attore primario specifica una chiave di ricerca; 2.2. il sistema ricerca la vendita specificata nel database; 2.3. <i>if</i> la vendita specificata è presente nel database 2.3.1. il sistema visualizza i corrispettivi dati; 2.4. <i>else</i> 2.4.1. il sistema visualizza un messaggio di errore; 3. <i>else if</i> l'attore primario vuole aggiornare i dati di una vendita 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca la vendita specificata nel database; 3.3. <i>if</i> la vendita specificata è presente nel database 3.3.1. l'attore primario specifica i nuovi dati da inserire; 3.3.2. il sistema procede con l'aggiornamento dei dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole rimuovere i dati di una vendita 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca la vendita specificata nel database; 4.3. <i>if</i> la vendita specificata è presente nel database 4.3.1. il sistema procede con la sua rimozione; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: ADDVendita
ID: CU10
Breve descrizione: Questo caso d'uso consente l'inserimento di una vendita da parte dell'amministratore.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole inserire la vendita di un libro; 2. l'attore primario inserisce i dati chiave del libro da vendere; 3. Il sistema verifica la disponibilità del libro richiesto, in particolare: 3.1. il sistema verifica che il libro sia presente; 3.2. il sistema verifica che la quantità corrente del libro sia diversa da 0; 4. <i>if</i> il libro ricercato non è presente (quantità corrente è uguale a 0) 4.1. il sistema invia un opportuno messaggio; 5. <i>else</i> 5.1. il sistema preleva tutte le informazioni relative al libro prescelto; 5.2. <i>if</i> il cliente persona o associazione è già tracciato nel database 5.3.1. il sistema preleva tutti i dati relativi al cliente persona o associazione; 5.3. <i>else</i> 5.3.1. l'attore primario inserisce tutti i dati relativi al cliente persona o associazione che sta effettuando l'acquisto; 5.3.2. il sistema memorizza nel database i dati inseriti; 5.4. il sistema registra tutti i dati relativi alla vendita; 5.5. il sistema crea un nuovo documento fiscale; 5.6. il sistema aggiorna la quantità corrente del libro;
Postcondizioni: Il libro è stato venduto se e solo se aveva una quantità corrente diversa da 0. Il sistema ha aggiornato la quantità corrente del libro.
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: CRUDDocFis
ID: CU11
Breve descrizione: Questo caso d'uso consente l'inserimento, la rimozione, la modifica e la ricerca di informazioni relative ad un documento fiscale.
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando l'attore primario vuole effettuare un'operazione CRUD sui dati relativi ad un documento fiscale; 2. <i>if</i> l'attore primario vuole inserire un nuovo documento fiscale 2.1. l'attore primario inserisce i dati relativi al nuovo documento fiscale; 2.2. il sistema memorizza tali dati nel database; 3. <i>else if</i> l'attore primario vuole ricercare nel database i dati di un documento fiscale 3.1. l'attore primario specifica una chiave di ricerca; 3.2. il sistema ricerca il documento fiscale specificato nel database; 3.3. <i>if</i> il documento fiscale specificato è presente nel database 3.3.1. il sistema visualizza i corrispettivi dati; 3.4. <i>else</i> 3.4.1. il sistema visualizza un messaggio di errore; 4. <i>else if</i> l'attore primario vuole aggiornare i dati di un documento fiscale 4.1. l'attore primario specifica una chiave di ricerca; 4.2. il sistema ricerca il documento fiscale specificato nel database; 4.3. <i>if</i> il documento fiscale specificato è presente nel database 4.3.1. l'attore primario specifica i nuovi dati da inserire; 4.3.2. il sistema procede con l'aggiornamento dei dati; 4.4. <i>else</i> 4.4.1. il sistema visualizza un messaggio di errore; 5. <i>else if</i> l'attore primario vuole rimuovere i dati di un documento fiscale 5.1. l'attore primario specifica una chiave di ricerca; 5.2. il sistema ricerca il documento fiscale specificato nel database; 5.3. <i>if</i> il documento fiscale specificato è presente nel database 5.3.1. il sistema procede con la sua rimozione; 5.4. <i>else</i> 5.4.1. il sistema visualizza un messaggio di errore;
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: RicercaLibro
ID: CU12
Breve descrizione: Questo caso d'uso consente ad un cliente la ricerca di informazioni relative ad un libro.
Attori primari: Cliente
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario <u>non</u> deve disporre di un account di utilizzo con appropriati diritti di accesso; la ricerca dei libri disponibili è aperta a tutti (il cliente <u>non</u> deve essere tesserato).
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando il cliente vuole effettuare un'operazione di ricerca sui dati relativi ad un libro; 2. Include(ResearchLibro); //questo caso d'uso non è altro che la R di CRUDLibro
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: CompraLibro
ID: CU13
Breve descrizione: Questo caso d'uso consente ad un cliente persona o associazione di acquistare un libro.
Attori primari: Cliente
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario <u>non</u> deve disporre di un account di utilizzo con appropriati diritti di accesso (il cliente <u>non</u> deve essere tesserato).
Sequenza degli eventi principale: 1. il caso d'uso inizia quando il cliente persona o associazione vuole acquistare un libro; 2. Include(AddVendita);
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: Nessuna

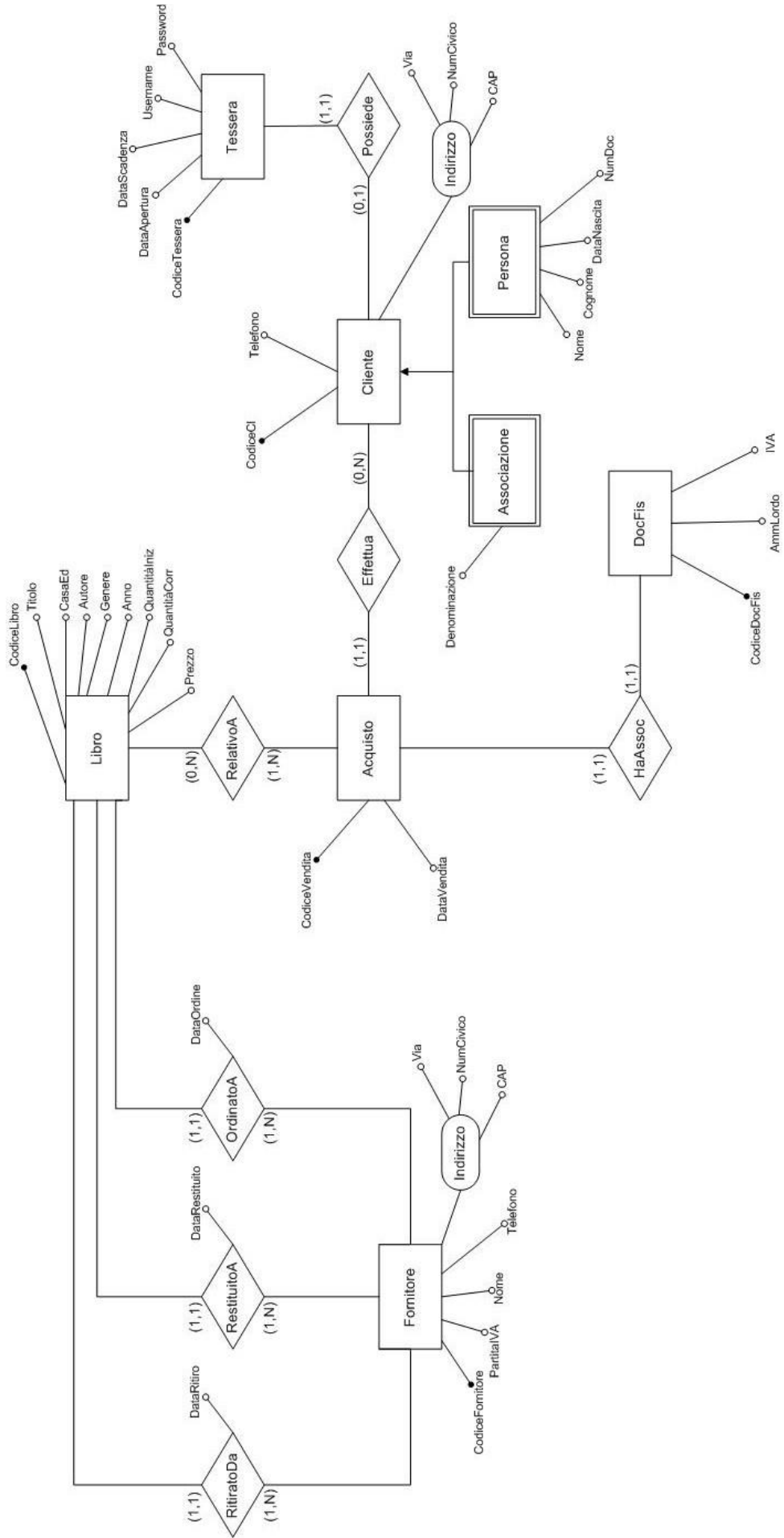
Caso d'uso: ReadOwnDetails
ID: CU14
Breve descrizione: Questo caso d'uso consente ad un cliente persona o associazione tesserato la lettura dei propri dati.
Attori primari: Cliente
Attori secondari: Nessuno
Precondizioni: 1. L'attore primario deve disporre di un account di utilizzo con appropriati diritti di accesso (il cliente deve essere tesserato).
Sequenza degli eventi principale: 1. il caso d'uso inizia quando il cliente persona o associazione tesserato vuole effettuare una operazione di ricerca sui propri dati; 2. <code>Include(ResearchTessera); // non è altro che la R di CRUDTessera</code>
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: Nessuna

1.4 Progettazione Concettuale

Lo schema E/R è corredato da una documentazione di supporto costituita da:

- dizionario delle entità
- dizionario delle relazioni
- dizionario dei vincoli

Lo schema concettuale per il sistema informativo in esame è riportato nella figura seguente:



Analisi delle cardinalità:

- L'entità **Libro** è identificata da CodiceLibro, che rappresenta la chiave primaria, ed è inoltre caratterizzata da Titolo, CasaEditrice, Autore, Genere, Anno, QuantitàIniziale, QuantitàCorrente e Prezzo. L'attributo QuantitàCorrente viene usato per calcolare il numero di copie vendute anche se questo dato potrebbe essere ricavato implementando un metodo Count, che però richiede molte operazioni. L'entità **Libro** ha cardinalità (0,N) nella relazione **RelativoA** perché un libro può essere acquistato (una o più volte) o meno; nelle relazioni **RitiratoDa**, **RestituitoA** e **OrdinatoA** ha invece cardinalità (1,1) in quanto deve obbligatoriamente essere ritirato, restituito o ordinato da un fornitore (generico);
- L'entità **Acquisto** è identificata da CodiceVendita, che rappresenta la chiave primaria, ed è inoltre caratterizzata da un attributo DataVendita. Ha cardinalità (1,1) nella relazione **HaAssoc**, perché ogni acquisto ha associato un documento fiscale, mentre la relazione **RelativoA** ha cardinalità (1,N) perché l'acquisto può essere relativo contemporaneamente ad uno o più libri;
- l'entità **DocFis** ha cardinalità (1,1) nella relazione **HaAssoc** poiché ogni documento fiscale è associato ad un acquisto;
- L'entità **Cliente** è identificata da CodiceCl che è la chiave primaria ed è caratterizzata dall'attributo Telefono. Un Cliente può essere Persona o Associazione. Nel caso sia una Persona avrà come attributi Nome, Cognome, DataNascita e NumDoc mentre nel caso sia un'Associazione avrà come attributo Denominazione. L'entità **Cliente** ha cardinalità (0,1) nella relazione **Possiede** perché un cliente può avere 0 o 1 tessera mentre ha cardinalità (0,N) nella relazione **Effettua** perché un cliente può fare 0 o N acquisti;
- L'entità **Tessera** è identificata da CodiceTessera che è la chiave primaria ed è inoltre caratterizzata da DataApertura, DataScadenza, Username e Password.

Questa entità ha cardinalità (1,1) nella relazione **Possiede** perché una tessera può essere posseduta da uno e un solo cliente;

- L'entità **Fornitore** è identificata da un CodiceFornitore che è la chiave primaria ed è inoltre caratterizzata da PartitaIVA, Nome, Telefono e dall'attributo composto Indirizzo. Questa entità ha cardinalità (1,N) nelle relazioni **RitiratoDa**, **RestituitoA** e **OrdinatoA** perché ad un fornitore può essere richiesto il ritiro, la restituzione o l'ordine di uno o più libri contemporaneamente;
- **OrdinatoA**: associa al relativo fornitore tutte le copie ordinate di uno stesso libro; il numero di copie ordinate di un libro corrisponde alla quantità iniziale della relativa istanza dell'entità Libro istanziata al momento dell'ordine quando ancora l'associata quantità corrente vale 0; per tale motivo, seppure una sua istanza possa avere una quantità iniziale maggiore di 1, l'entità Libro coinvolta in OrdinatoA presenta cardinalità massima 1; inoltre tutte le istanze dell'entità Libro devono essere coinvolte in OrdinatoA, quindi anche la cardinalità minima sarà pari a 1;
- **RitiratoDa**: associa al relativo fornitore tutte le copie ritirate di uno stesso libro ordinate in precedenza; al momento del ritiro delle copie ordinate di un libro, la quantità corrente della relativa istanza dell'entità Libro passa da 0 al valore della quantità iniziale; per tale motivo, seppure una sua istanza possa avere una quantità iniziale maggiore di 1, l'entità Libro coinvolta in RitiratoDa presenta cardinalità massima 1; inoltre per lo stesso motivo visto al punto precedente, tutte le istanze dell'entità Libro devono essere coinvolte in RitiratoDa, quindi anche la cardinalità minima sarà pari a 1;
- **RestituitoA**: associa al relativo fornitore tutte le copie vendibili di un libro rimaste invendute (cioè un numero di copie pari alla quantità corrente della relativa istanza dell'entità Libro) e per tale motivo restituite; a restituzione avvenuta, la quantità corrente della relativa istanza dell'entità Libro passa a 0;

per quanto detto sopra l'entità Libro coinvolta in RestituitoA presenta cardinalità massima 1 e, come nei casi precedenti, tutte le istanze dell'entità Libro devono essere coinvolte in RestituitoA, quindi anche la cardinalità minima sarà pari a 1;

Dizionario delle Entità:

Entità	Descrizione	Attributi	Identificatore
Cliente	Padre della generalizzazione su un cliente della libreria	CodiceCl, Telefono, Indirizzo	CodiceCl
Persona	Figlio di Cliente, di tipo persona fisica	Nome, Cognome, DataNascita, NumDoc	CodiceCl
Associazione	Figlio di Cliente, di tipo associazione culturale	Denominazione	CodiceCl
DocFis	Documento fiscale relativo ad un acquisto	CodiceDocFis, AmmLordo, IVA, AmmontareNetto	CodiceDocFis
Tessera	Tessera di un cliente tesserato	CodiceTessera, DataApertura, DataScadenza, UserName, Password	CodiceTessera
Acquisto	Acquisto di un libro da parte di un cliente	CodiceVendita, DataVendita	CodiceVendita
Libro	Libri in vendita nella libreria	CodiceLibro, Titolo, CasaEd, Autore, Genere, Anno, QuantitàIniz, QuantitàCorr, Prezzo	CodiceLibro
Fornitore	Fornitore della libreria	CodiceFornitore, PartitaIVA, Nome, Telefono, Indirizzo	CodiceFornitore

Dizionario delle Relazioni:

Relazione	Descrizione	Entità coinvolte	Attributi
Possiede	Associa un cliente persona o associazione ad una tessera	Cliente (0,1), Tessera (1,1)	-
Effettua	Associa l'acquisto di un libro ad un cliente	Cliente (0,N), Acquisto (1,1)	-
HaAssoc	Associa un documento fiscale ad un acquisto	Acquisto (1,1), DocFis (1,1)	-
RelativoA	Associa il libro acquistato ad uno o più acquisti	Acquisto (1,N), Libro (0,N)	-
RitiratoDa	Associa il fornitore del libro ad un libro ritirato	Libro (1,1), Fornitore (1,N)	DataRitiro
RestituitoA	Associa il fornitore del libro ad un libro restituito	Libro (1,1), Fornitore (1,N)	DataRestituito
OrdinatoA	Associa il fornitore del libro ad un libro ordinato	Libro (1,1), Fornitore (1,N)	DataOrdine

Dizionario dei Vincoli:

Ident. Regola	Regola
A1	Se la libreria restituisce ad un fornitore un libro, restituisce tutte le copie rimaste invendute pari alla quantità iniziale meno le copie vendute.
A2	La password di una tessera deve essere almeno di otto caratteri (e al più quindici caratteri) di cui almeno due alfanumerici per i clienti tesserati e l'amministratore.
A3	Un cliente può effettuare un acquisto di un libro solo se la sua quantità corrente è diversa da 0.

1.5 Progettazione Logica

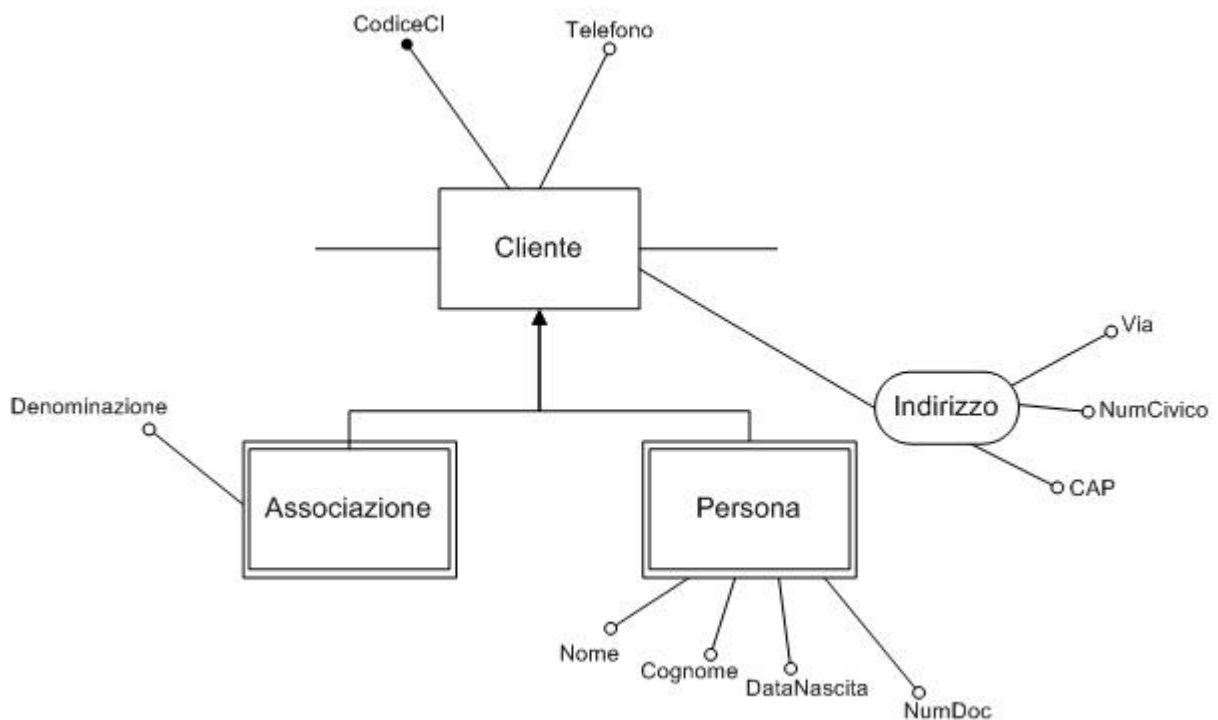
- **Ristrutturazione dello Schema E/R**

1) Analisi delle ridondanze

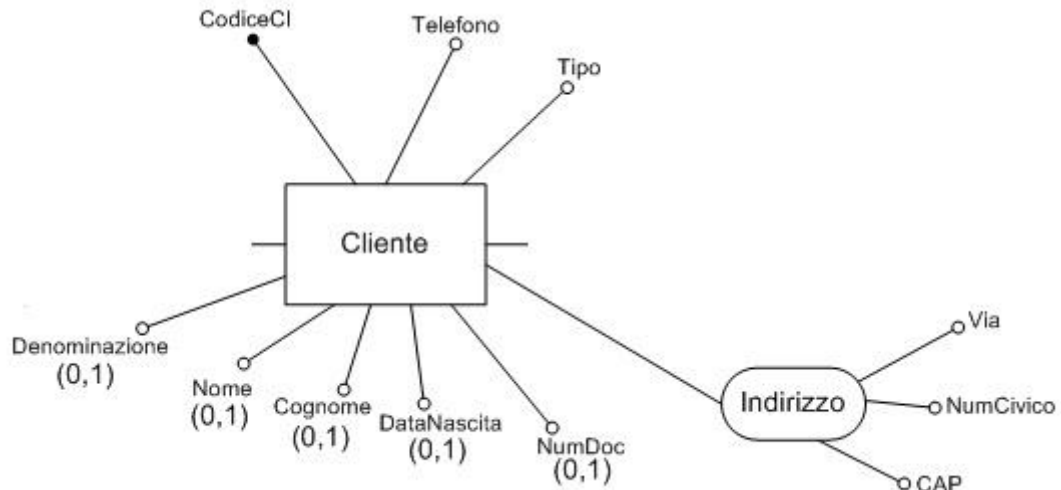
In presenza di uno schema E/R progettato bene, non è necessario attuare l'analisi delle ridondanze.

2) Eliminazione delle generalizzazioni

Nello schema E/R vi è una generalizzazione da eliminare tra Cliente, Persona e Associazione:



visto che il padre è coinvolto in 2 relazioni mentre i figli in nessuna, si ritiene che il padre sia più importante dei figli (si è scelto come indicatore di importanza il numero di relazioni in cui le entità sono coinvolte) e si procede accorpando i figli nel padre (si spostano tutti gli attributi e le relazioni dei figli sul padre aggiornando opportunamente le cardinalità e aggiungendo un attributo Tipo per mantenere memoria dei figli):



3) Partizionamento/Accorpamento di Entità e Associazioni

Per lo schema E/R proposto, non si ritiene necessario partizionare o accorpare le entità.

4) Eliminazione degli attributi multivalore

Nello schema E/R proposto, non sono presenti attributi multivalore.

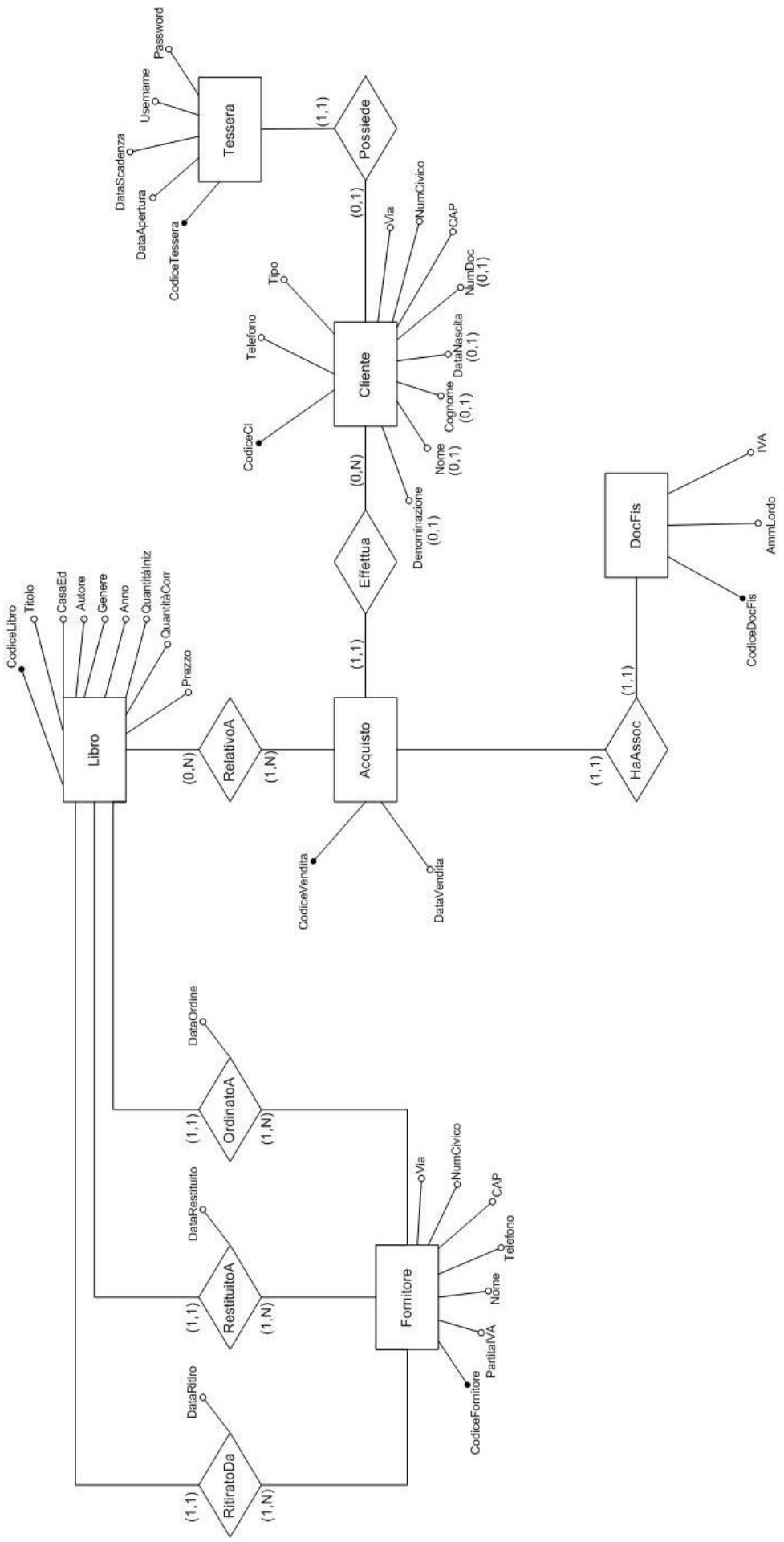
5) Eliminazione degli attributi composti

L'unico attributo composto presente nello schema E/R è Indirizzo in Cliente e Fornitore; la ristrutturazione prevede semplicemente di sostituire all'attributo composto gli stessi attributi semplici che lo compongono.

6) Scelta degli identificatori principali

Le chiavi primarie sono già tutte ben definite e non potranno assumere valori nulli. Sono tutte chiavi primarie semplici (costituite cioè da un singolo attributo) e tutte costituite da attributi interni.

Lo schema E/R ristrutturato che si ottiene a seguito di queste modifiche, risulta il seguente:



- **Traduzione dello schema E/R ristrutturato verso il modello relazionale**

A partire da uno schema E/R ristrutturato si costruisce uno schema relazionale. Si ottiene il seguente schema relazionale (si è scelto di chiamare con lo stesso nome gli attributi coinvolti in un vincolo di integrità referenziale per maggiore chiarezza, in modo che i collegamenti risultino di immediata lettura):

- CLIENTE (CodiceCl, Tipo, Telefono, Via, NumCivico, CAP, NumDoc, DataNascita, Nome, Cognome, Denominazione)
- TESSERA (CodiceTessera, DataApertura, DataScadenza, Username, Password, CodiceCl)
- DOCFIS (CodiceDocFis, AmmLordo, IVA)
- ACQUISTO (CodiceVendita, DataVendita, CodiceCl, CodiceLibro, CodiceDocFis)
- LIBRO (CodiceLibro, Titolo, CasaEd, Autore, Genere, Anno, QuantitàIniz, QuantitàCorr, Prezzo)
- ORDINE (CodiceLibro, CodiceFornitore, DataOrdine)
- RITIRO (CodiceLibro, CodiceFornitore, DataRitiro)
- RESTITUZIONE (CodiceLibro, CodiceFornitore, DataRestituito)
- FORNITORE (CodiceFornitore, PartitaIVA, Nome, Telefono, Via, NumCivico, CAP)

Nella tabella CLIENTE, visto che gli attributi Nome e Cognome saranno vuoti (NULL) nel caso in cui Tipo = 'Associazione', mentre Denominazione sarà vuoto (NULL) nel caso in cui Tipo = 'Persona', cioè la coppia Nome e Cognome è alternativa alla Denominazione, possiamo sostituirli con un unico attributo *Nome*; se Tipo = "Persona" questo attributo rappresenterà Nome e Cognome (nel medesimo campo); se Tipo = "Associazione" questo attributo rappresenterà la Denominazione.

Per un'associazione, non esistendo valori per gli attributi NumDoc e DataNascita, gli stessi verranno posti a NULL. Questo non rappresenta un grosso problema in quanto si prevede che il numero totale di associazioni che acquistano libri presso la libreria in un intero anno sia nettamente inferiore al numero di clienti persona, quindi il totale dei campi NULL sarà accettabile.

Per cui l'ottimizzazione risultante è la seguente:

CLIENTE (CodiceCl, Tipo, Telefono, Via, NumCivico, CAP, NumDoc, DataNascita, Nome)

A seguito di questa modifica si ottiene il seguente **schema logico definitivo**:

- CLIENTE (CodiceCl, Tipo, Telefono, Via, NumCivico, CAP, NumDoc, DataNascita, Nome)
- TESSERA (CodiceTessera, DataApertura, DataScadenza, Username, Password, CodiceCl)
- DOCFIS (CodiceDocFis, AmmLordo, IVA)
- ACQUISTO (CodiceVendita, DataVendita, CodiceCl, CodiceLibro, CodiceDocFis)
- LIBRO (CodiceLibro, Titolo, CasaEd, Autore, Genere, Anno, QuantitàIniz, QuantitàCorr, Prezzo)
- ORDINE (CodiceLibro, CodiceFornitore, DataOrdine)
- RITIRO (CodiceLibro, CodiceFornitore, DataRitiro)
- RESTITUZIONE (CodiceLibro, CodiceFornitore, DataRestituito)
- FORNITORE (CodiceFornitore, PartitaIVA, Nome, Telefono, Via, NumCivico, CAP)

Vincoli d'integrità referenziale:

Attributo referente (vincolato)	Attributo riferito (vincolante)
Tessera.CodiceCl	Cliente.CodiceCl
Acquisto.CodiceCl	Cliente.CodiceCl
Acquisto.CodiceDocFis	DocFis.CodiceDocFis
Acquisto.CodiceLibro	Libro.CodiceLibro
Ordine.CodiceLibro	Libro.CodiceLibro
Ordine.CodiceFornitore	Fornitore.CodiceFornitore
Ritiro.CodiceLibro	Libro.CodiceLibro
Ritiro.CodiceFornitore	Fornitore.CodiceFornitore
Restituzione.CodiceLibro	Libro.CodiceLibro
Restituzione.CodiceFornitore	Fornitore.CodiceFornitore

1.6 Progettazione Fisica

- **Adattamento dello schema logico al DBMS prescelto:**

Questo passo non è necessario in quanto il DBMS scelto è Oracle.

- **Analisi delle transazioni:**

Sono state individuate le seguenti transazioni:

- Inserimento di un libro
- Ricerca o rimozione di un libro
- Inserimento di un cliente
- Ricerca di un cliente tesserato
- Aggiornamento dei dati di un cliente
- Inserimento di un fornitore
- Visualizzare tutti i libri acquistati

Utilizzando i simboli: I per inserimento, R per la lettura, U per l'aggiornamento e D per la cancellazione, la corrispondente matrice dei riferimenti incrociati transazioni/relazioni è la seguente:

	A				B				C				D				E				F				G																							
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D																
Libro	x					x		X																																								
Cliente									x					x				x	x																													
Tessera														x																																		
Fornitore																								x																								
Acquisto																																																
DocFis																																																

Analizzando la matrice si nota che sia la relazione Libro che la relazione Cliente vengono accedute da 3 delle 7 transazioni, pertanto è richiesto un accesso efficiente a tali transazioni per evitare problemi di performance.

Ci sono operazioni di aggiornamento che coinvolgono gli attributi: Telefono, Via, NumCivico e CAP di Cliente.

Ci sono operazioni di ricerca che coinvolgono gli attributi: CodiceLibro, Titolo, Autore, CasaEd e Anno di Libro, mentre: CodiceTessera e DataScadenza di Tessera.

La transazione G si prevede venga eseguita ogni sera; mentre le altre non presentano un'assiduità regolare.

- **Scelta dell'organizzazione dei file:**

Si preferisce usare le tabelle Hash perché si prevede che la maggior parte delle query restituiscono un solo risultato.

- **Scelta degli indici:**

Oltre agli indici sulle chiavi primarie già automaticamente inseriti dal sistema appaiono utili altri indici sulle chiavi secondarie. Un approccio per determinare quali indici secondari sono necessari consiste nel produrre una wish-list.

Wish-List:

- Nella tabella TESSERA appare utile un indice associato all'attributo CodiceCl essendo un attributo di join ereditato dalla tabella CLIENTE;
- Nella tabella ACQUISTO appare utile un indice sull'attributo CodiceCl essendo un attributo di join con la tabella CLIENTE e un indice sull'attributo CodiceLibro essendo un attributo di join con la tabella LIBRO e un indice sull'attributo CodiceDocFis essendo un attributo di join con la tabella DOCFIS;
- Nella tabella LIBRO appare utile un indice sull'attributo Titolo, su Autore e su CasaEd;

- Nella tabella ORDINE appare utile un indice associato alla coppia <DataOrdine, CodiceFornitore> (mettendo prima DataOrdine per avere automaticamente incluso un indice secondario anche sul solo primo attributo) per la ricerca immediata degli ordini;
 - Nella tabella RITIRO appare utile un indice associato alla coppia <DataRitiro, CodiceFornitore> (mettendo prima DataRitiro per avere automaticamente incluso un indice secondario anche sul solo primo attributo) per la ricerca immediata dei ritiri;
 - Nella tabella RESTITUZIONE appare utile un indice associato alla coppia <DataRestituito, CodiceFornitore> (mettendo prima DataRestituito per avere automaticamente incluso un indice secondario anche sul solo primo attributo) per la ricerca immediata delle restituzioni;
- **Stima delle richieste di spazio su disco:**

Per memorizzare una tupla della tabella CLIENTE appare necessaria la seguente quantità di memoria:

 - 4 byte per CodiceCl
 - 3 byte per Tipo (solo se si usano abbreviazioni a 3 caratteri come Tipo = “Ass” e Tipo = “Per”)
 - 15 byte per Telefono
 - 50 byte per Via
 - 5 byte per NumCivico
 - 5 byte per CAP
 - 20 byte per NumDoc
 - 10 byte per DataNascita
 - 50 byte per Nome

Di conseguenza il totale sarà: 162 byte. Se si stima che in un anno vengono memorizzati 8.200 clienti saranno necessari 1.328.400 byte ovvero circa 1.3 MB. In dieci anni 13.284.000 byte ovvero circa 13 MB.

Per memorizzare una tupla della tabella TESSERA appare necessaria la seguente quantità di memoria:

- 4 byte per CodiceTessera
- 10 byte per DataApertura
- 10 byte per DataScadenza
- 15 byte per Username
- 15 byte per Password
- 4 byte per CodiceCI

Di conseguenza il totale sarà: 58 byte. Se si stima che in un anno vengono memorizzati 7.000 tessere saranno necessari 406.000 byte ovvero circa 0.4 MB. In dieci anni 4.060.000 byte ovvero circa 4 MB.

Per memorizzare una tupla della tabella DOCFIS appare necessaria la seguente quantità di memoria:

- 4 byte per CodiceDocFis
- 7 byte per AmmLordo
- 4 byte per IVA

Di conseguenza il totale sarà: 15 byte. Se si stima che in un anno vengono memorizzati 49.000 Docfis saranno necessari 735.000 byte ovvero circa 0.7 MB. In dieci anni 7.350.000 byte ovvero circa 7 MB.

Per memorizzare una tupla della tabella ACQUISTO appare necessaria la seguente quantità di memoria:

- 4 byte per CodiceVendita
- 10 byte per DataVendita
- 4 byte per CodiceCI
- 4 byte per CodiceLibro
- 4 byte per CodiceDocFis

Di conseguenza il totale sarà: 26 byte. Se si stima che in un anno vengono memorizzati 216.000 acquisti saranno necessari 5.616.000 byte ovvero circa 5.5 MB. In dieci anni 56.160.000 byte ovvero circa 56 MB.

Per memorizzare una tupla della tabella LIBRO appare necessaria la seguente quantità di memoria:

- 4 byte per CodiceLibro
- 30 byte per Titolo
- 20 byte per CasaEd
- 30 byte per Autore
- 15 byte per Genere
- 4 byte per Anno
- 2 byte per QuantitàIniz
- 2 byte per QuantitàCorr
- 7 byte per Prezzo (come AmmLordo e IVA)

Di conseguenza il totale sarà: 114 byte. Se si stima che in un anno vengono memorizzati 24.000 libri saranno necessari 2.736.000 byte ovvero circa 2.7 MB. In dieci anni 27.360.000 byte ovvero circa 27 MB.

Per memorizzare una tupla della tabella ORDINE appare necessaria la seguente quantità di memoria:

- 4 byte per CodiceLibro

- 4 byte per CodiceFornitore
- 10 byte per DataOrdine

Di conseguenza il totale sarà: 18 byte. Se si stima che in un anno vengono memorizzati 9.600 ordini saranno necessari 172.800 byte ovvero circa 0.1 MB. In dieci anni 1.728.000 byte ovvero circa 1.7 MB.

Per memorizzare una tupla della tabella RITIRO appare necessaria la seguente quantità di memoria:

- 4 byte per CodiceLibro
- 4 byte per CodiceFornitore
- 10 byte per DataRitiro

Di conseguenza il totale sarà: 18 byte. Se si stima che in un anno vengono memorizzati 192.000 ritiri saranno necessari 3.456.000 byte ovvero circa 3.4 MB. In dieci anni 34.560.000 byte ovvero circa 34 MB.

Per memorizzare una tupla della tabella RESTITUZIONE appare necessaria la seguente quantità di memoria:

- 4 byte per CodiceLibro
- 4 byte per CodiceFornitore
- 10 byte per DataRestituito

Di conseguenza il totale sarà: 18 byte. Se si stima che in un anno vengono memorizzati 57.600 (30% dei ritiri) restituzioni di libri saranno necessari 1.036.800 byte ovvero circa 1 MB. In dieci anni 10.368.000 byte ovvero circa 10 MB.

Per memorizzare una tupla della tabella FORNITORE appare necessaria la seguente quantità di memoria:

- 4 byte per CodiceFornitore
- 11 byte per PartitaIVA (in Italia la partita IVA è costituita da 11 cifre)
- 30 byte per Nome
- 15 byte per Telefono
- 50 byte per Via
- 5 byte per NumCivico
- 5 byte per CAP

Di conseguenza il totale sarà: 120 byte. Se si stima che in un anno vengono memorizzati 60 Fornitori saranno necessari 7.200 byte ovvero circa 0.007 MB. In dieci anni 72.000 byte ovvero circa 0.07 MB.

Supponendo che il sistema proposto sia in esercizio per dieci anni, si stima che verranno quindi occupati un totale di 249.118.000 byte, ovvero circa 249 MB.

1.7 Definizione del diagramma delle classi

Si definiscono le classi associate a ciascuna tabella per il sistema di riferimento.

Una prima bozza della classe associata alle tabelle CLIENTE e TESSERA è la seguente:

class Cliente

Cliente

- cap
- codiceCI
- dataNascita
- nome
- numCivico
- numDoc
- telefono
- tipo
- via

- + dbInserisci()
- + dbModifica()
- + dbRicerca()
- + dbRimuovi()
- + getCap()
- + getCodiceCI()
- + getDataNascita()
- + getNome()
- + getNumCivico()
- + getNumDoc()
- + getTelefono()
- + getTipo()
- + getVia()
- + setCap()
- + setCodiceCI()
- + setDataNascita()
- + setNome()
- + setNumCivico()
- + setNumDoc()
- + setTelefono()
- + setTipo()
- + setVia()

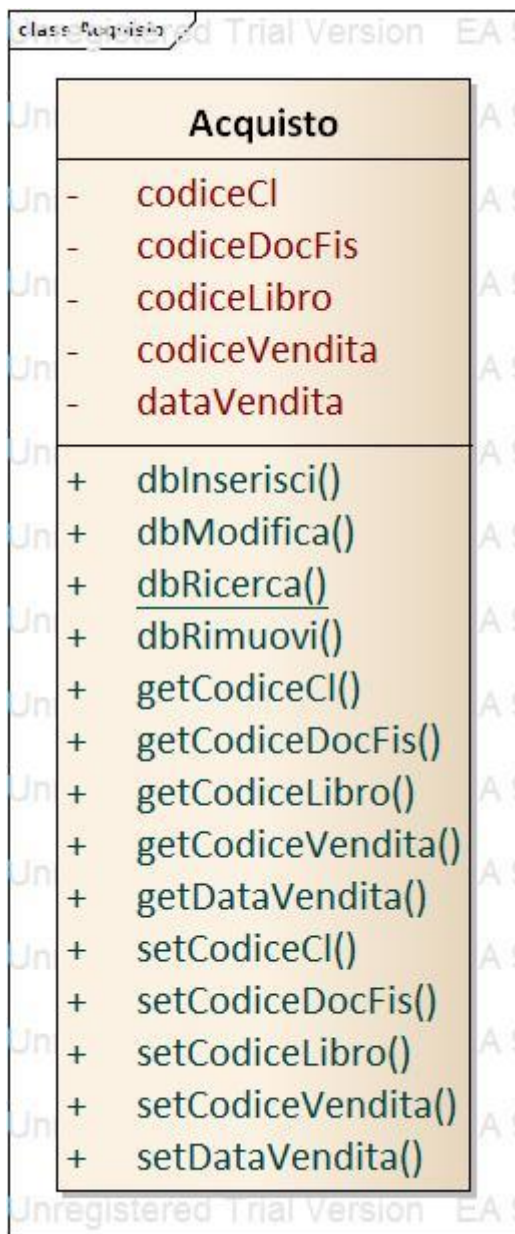
class Tessera

Tessera

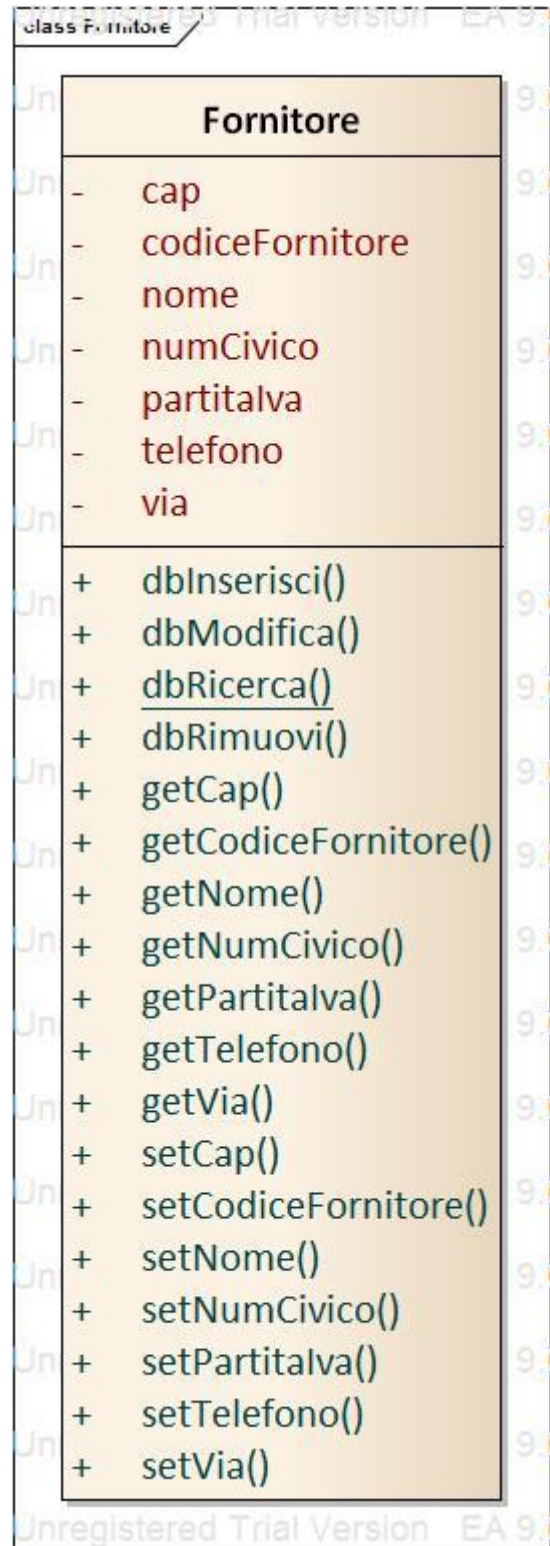
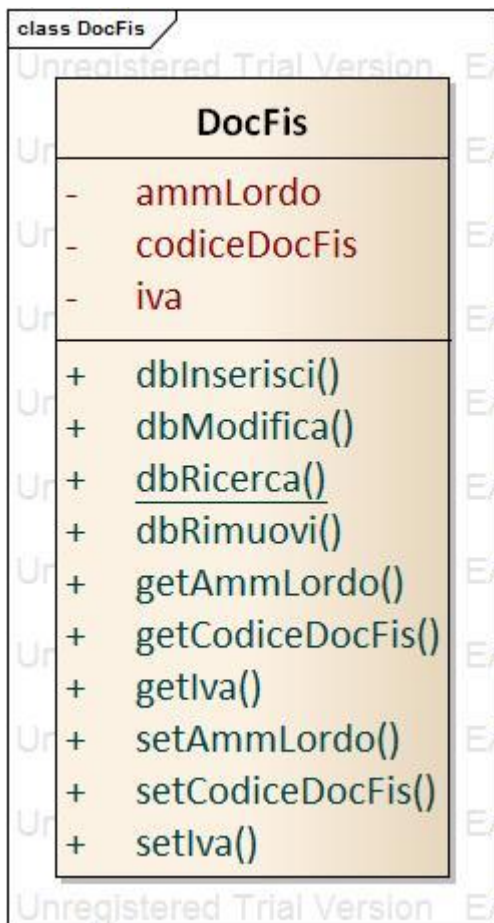
- codiceCI
- codiceTessera
- dataApertura
- dataScadenza
- password
- username

- + dbInserisci()
- + dbModifica()
- + dbRicerca()
- + dbRimuovi()
- + getCodiceCI()
- + getCodiceTessera()
- + getDataApertura()
- + getDataScadenza()
- + getPassword()
- + getUsername()
- + setCodiceCI()
- + setCodiceTessera()
- + setDataApertura()
- + setDataScadenza()
- + setPassword()
- + setUsername()

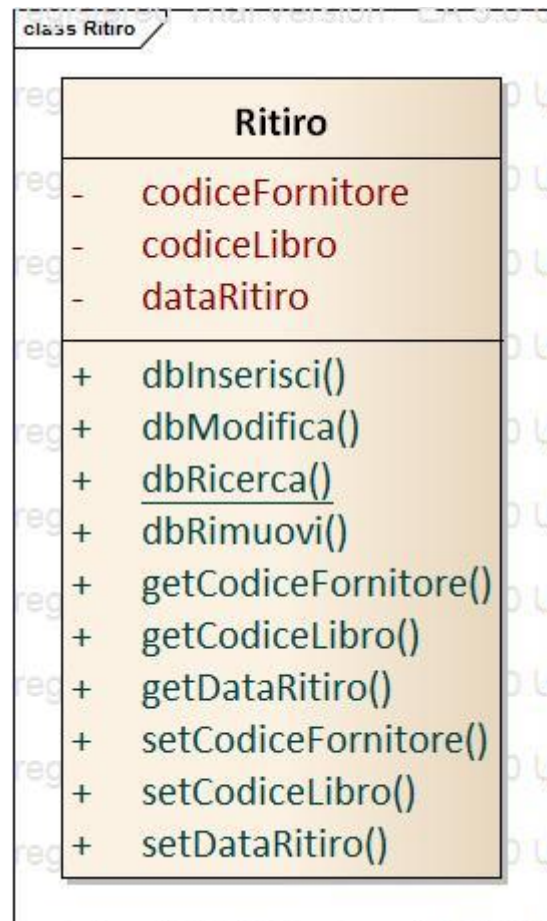
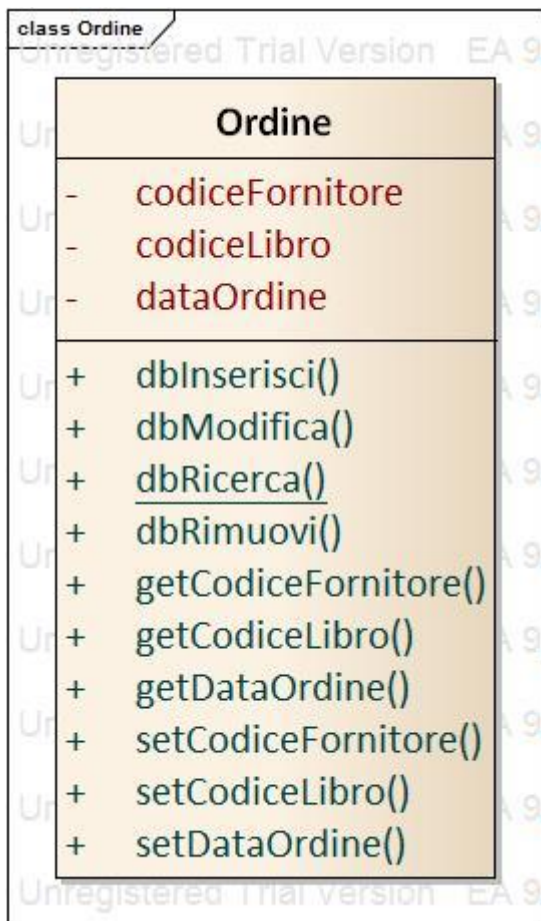
Una prima bozza della classe associata alle tabelle LIBRO e ACQUISTO è la seguente:



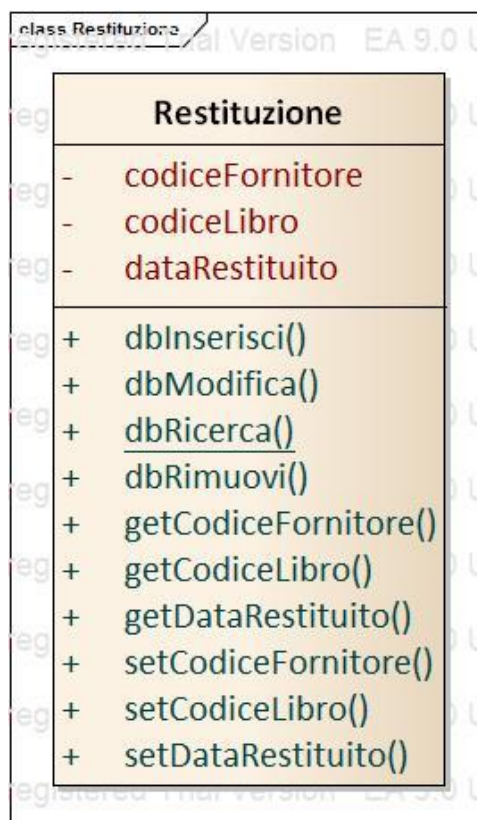
Una prima bozza della classe associata alle tabelle DOCFIS e FORNITORE è la seguente:



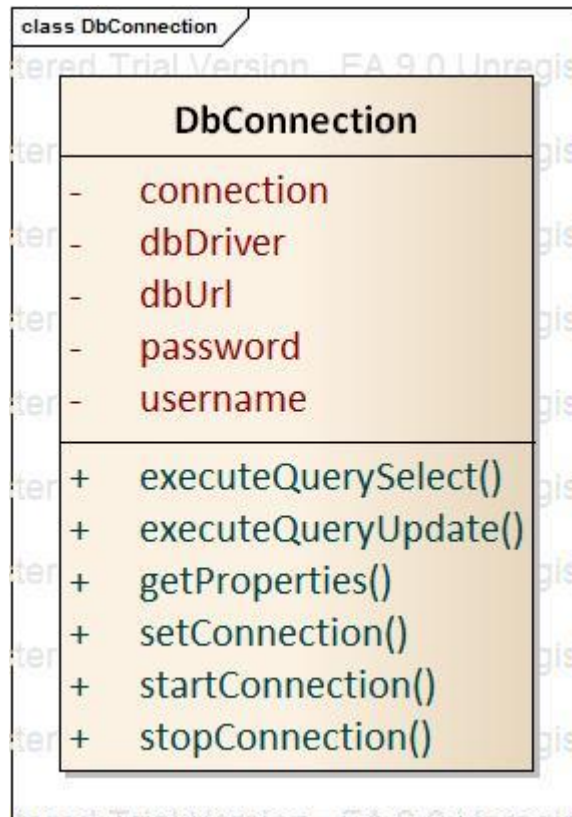
Una prima bozza della classe associata alle tabelle ORDINE e RITIRO è la seguente:



Una prima bozza della classe associata alla tabella RESTITUZIONE è la seguente:

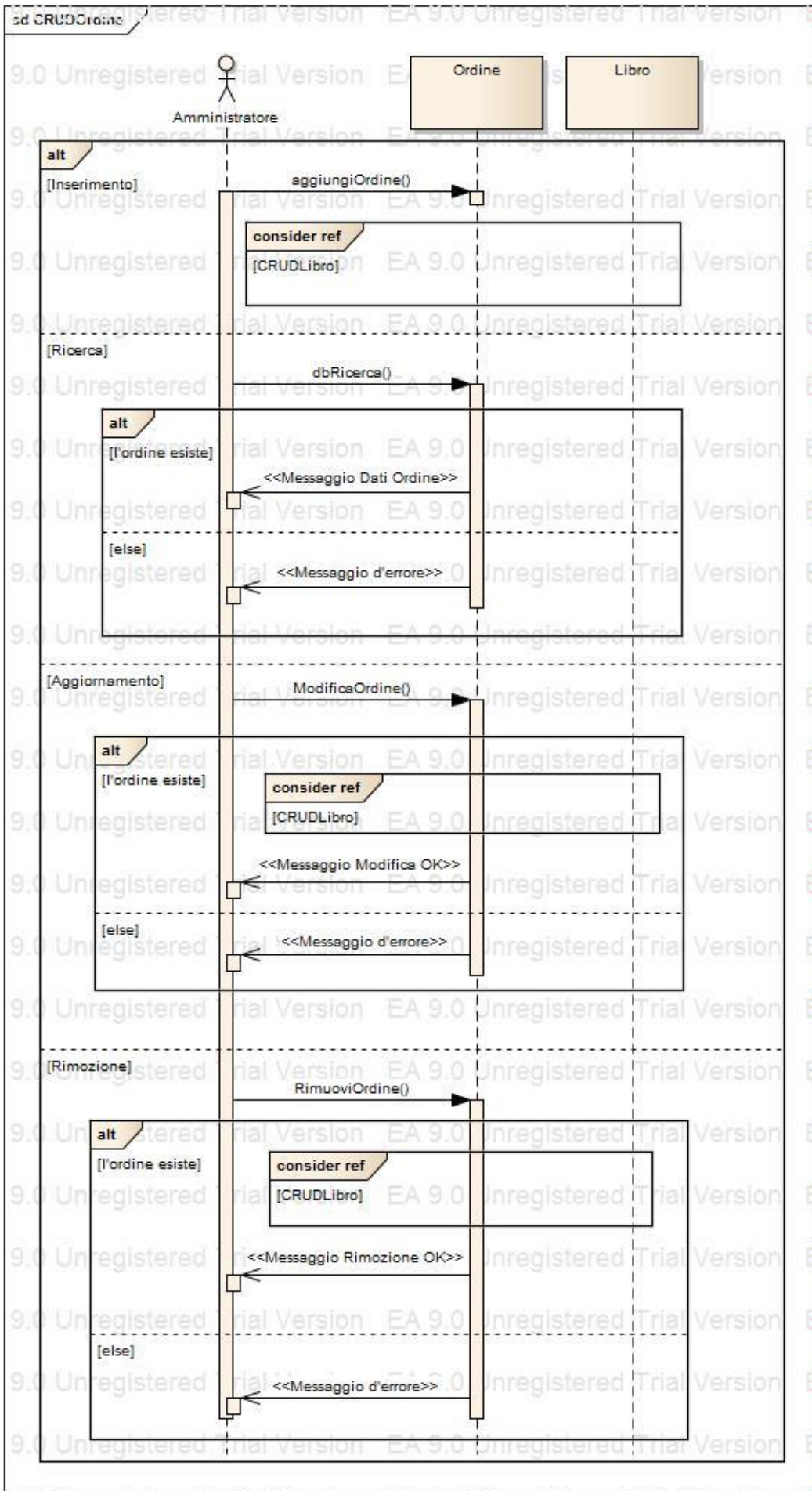


Oltre alle classi corrispondenti alle tabelle, si progetta un ulteriore **classe DBConnection** che si occupa della gestione, della connessione e della interrogazione al database sottostante. Questa classe ha lo scopo di fare da ponte tra il modello relazionale e il paradigma orientato agli oggetti.

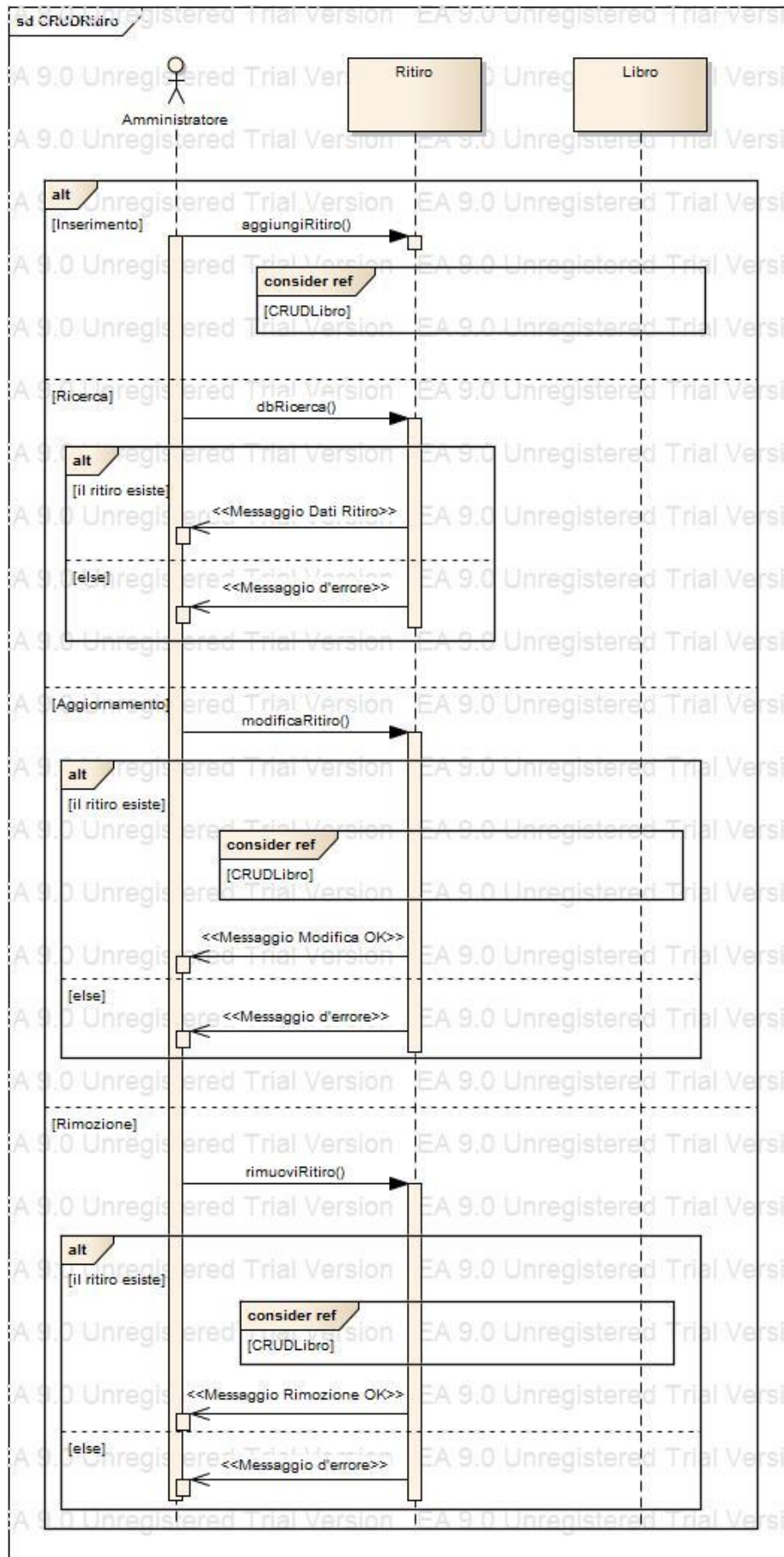


1.8 Progettazione delle funzionalità e raffinamento delle classi

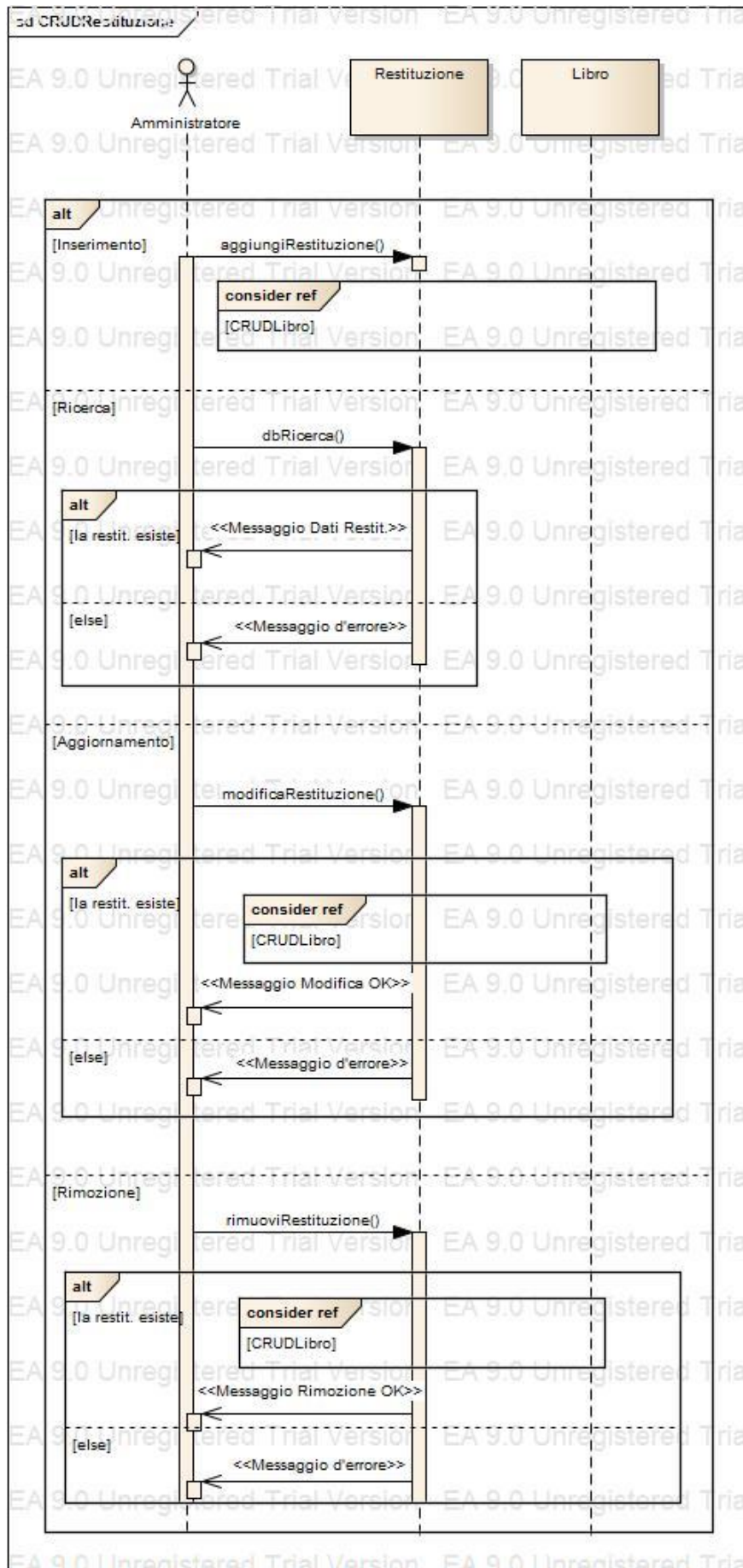
Il diagramma di sequenza relativo al caso d'uso CRUDOrdine è riportato nella seguente figura:



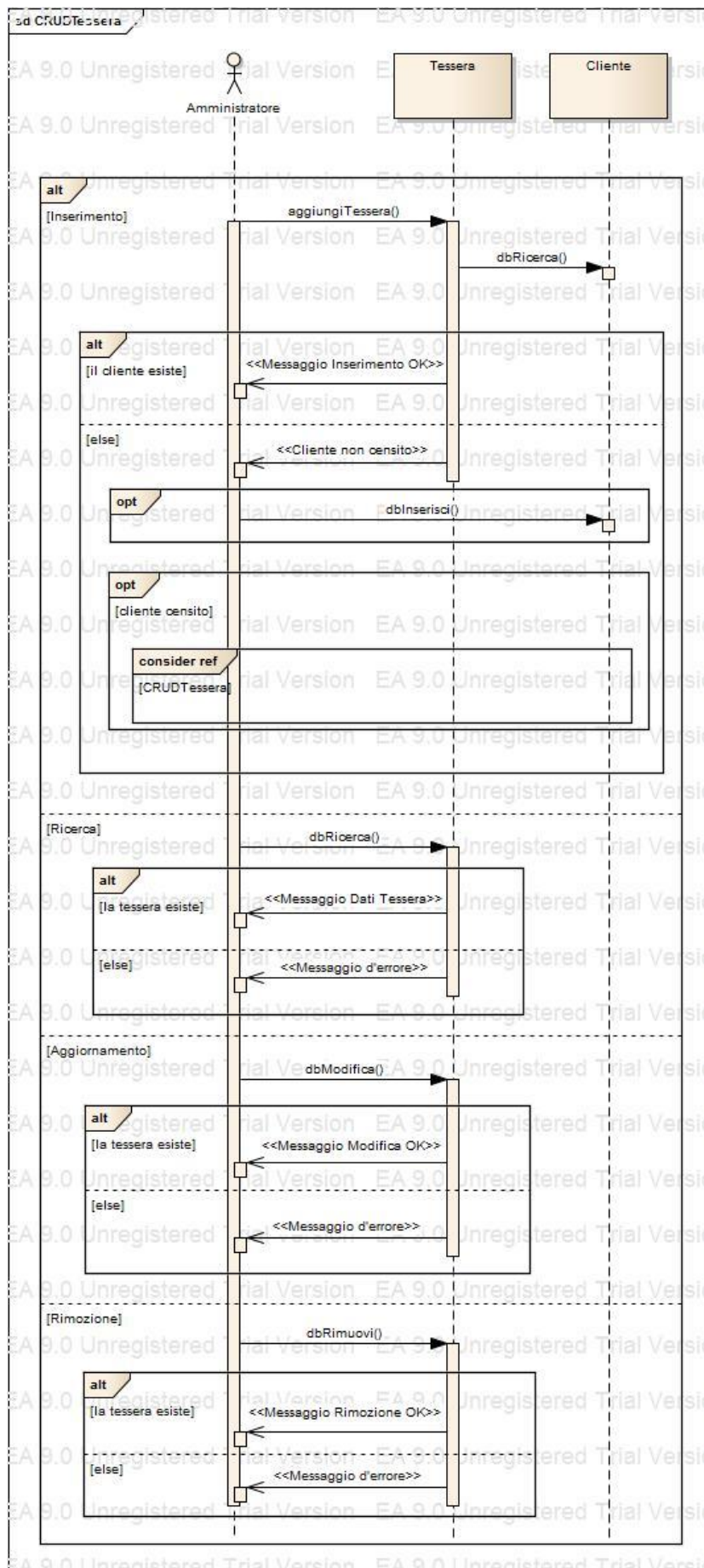
Il diagramma di sequenza relativo al caso d'uso CRUDRitiro è riportato nella seguente figura:



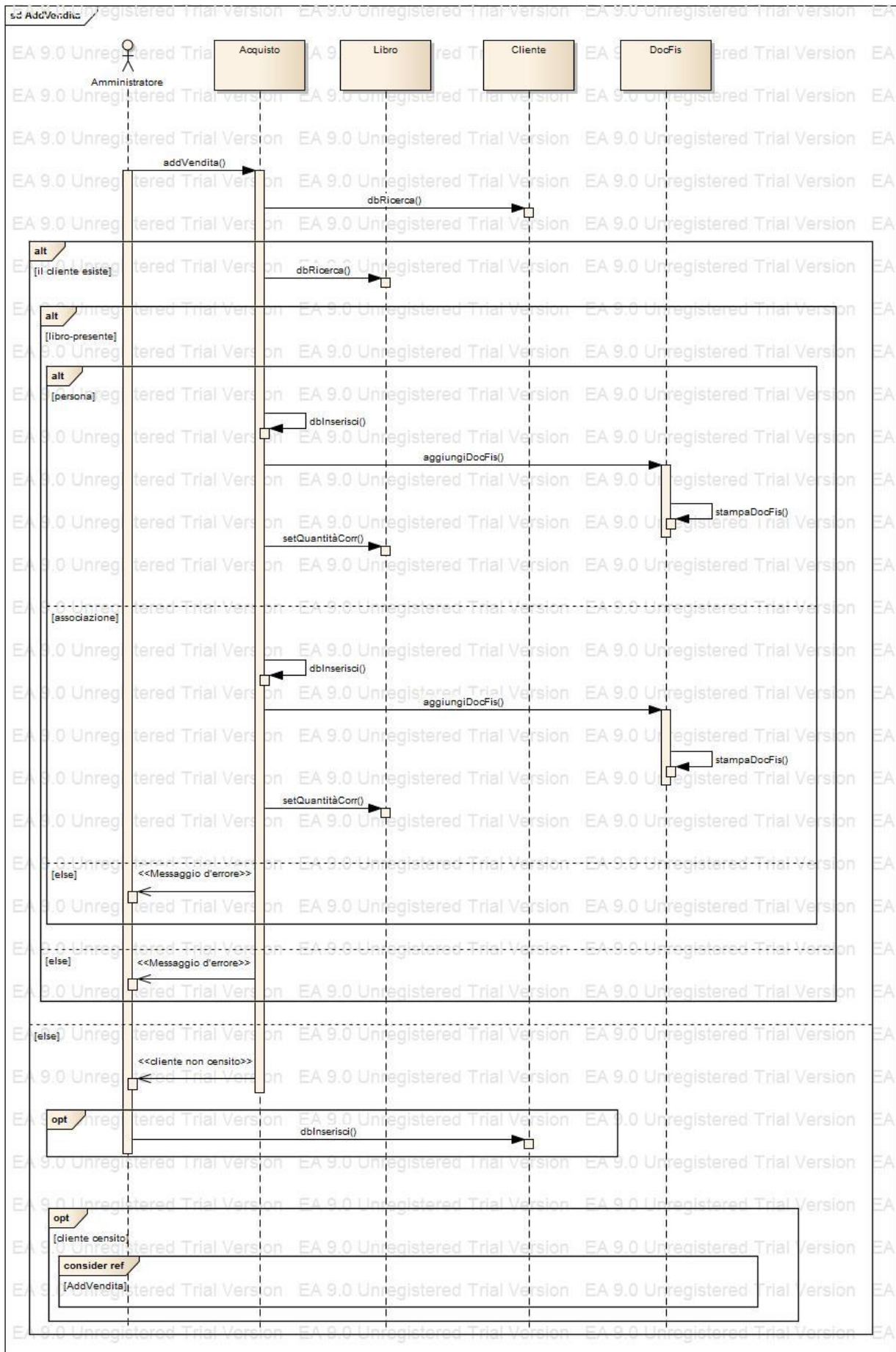
Il diag. di sequenza relativo al caso d'uso CRUDRestituzione è riportato nella seguente figura:



Il diagramma di sequenza relativo al caso d'uso CRUDTessera è riportato nella seguente figura:



Il diagramma di sequenza relativo al caso d'uso AddVendita è riportato nella seguente figura:



Esaminando i casi d'uso definiti in precedenza, possiamo notare come:

1. CRUDFornitore
2. CRUDClienteAssoc
3. CRUDClientePers
4. CRUDLibro
5. RUDVendita
6. CRUDDocFis

coinvolgono solo una classe e non richiedono elaborazioni sui dati pertanto non è ritenuto necessario costruire i corrispettivi diagrammi di sequenza.

Inoltre per i casi d'uso :

1. RicercaLibro
2. CompraLibro
3. ReadOwnDetails

il caso d'uso RicercaLibro non è altro che la R di CRUDLibro; il caso d'uso CompraLibro corrisponde al caso d'uso AddVendita dell'Amministratore; infine ReadOwnDetails non è altro che la R di CRUDTessera; esse pertanto sono attività implicitamente già sequenzializzate.

Dall'esame di questi diagrammi di sequenza si evince che è necessario aggiungere i seguenti metodi alle bozze di classi progettate in precedenza:

- aggiungiOrdine(), modificaOrdine() e rimuoviOrdine() alla classe Ordine (poiché i soli metodi CRUD interni non sono sufficienti, essendo necessarie attività su più tabelle);
- aggiungiRitiro(), modificaRitiro() e rimuoviRitiro() alla classe Ritiro (poiché i soli metodi CRUD interni non sono sufficienti, essendo necessarie attività su più tabelle);

- aggiungiRestituzione(), modificaRestituzione() e rimuoviRestituzione() alla classe Restituzione (poiché i soli metodi CRUD interni non sono sufficienti, essendo necessarie attività su più tabelle);
- aggiungiTessera() alla classe Tessera che chiami al suo interno dbRicerca() di Cliente;
- addVendita() alla classe Acquisto (il solo dbInserisci() è insufficiente);
- aggiungiDocFis() (poiché dbInserisci() da solo non eseguirebbe la stampa del DocFis) e stampaDocFis() alla classe DocFis;

Di seguito si riportano le classi raffinate con tutti i metodi aggiuntivi evinti dai diagrammi di sequenza ad esclusione delle classi come DBConnection che non hanno subito alcuna modifica rispetto alle precedenti bozze:

