

Università degli Studi di Urbino “Carlo Bo”

Corso di Laurea in Informatica Applicata

Relazione Di Progetto

INGEGNERIA DEL SOFTWARE

Prof. Delpriori – Prof. Mezzina

Software per la gestione della vendita online
di un'azienda metalmeccanica

Massimiliano Carrubba
Matricola 242526
A.A. 2019/2020

1. Specifiche del software

Il cliente richiede la realizzazione di un software di gestione degli ordini di una azienda metalmeccanica.

Ogni oggetto in vendita può essere di due tipologie:

meccanica pesante (barre, profilati, ...) oppure componentistica (viti, dadi, bulloni, ...).

Un ordine può comportare l'acquisto di uno o più oggetti, purché questi riguardino una per volta delle tipologie sopra citate.

Ogni cliente avrà assegnato un indice di gradimento commisurato alla quantità di commesse ed all'importo di merce acquistata.

Ogni commessa può essere formulata sia da un cliente associato sia da un nuovo cliente: quest'ultimo diverrà automaticamente un cliente associato al termine del pagamento del primo ordine.

Le coordinate bancarie sono obbligatorie per ogni nuovo cliente, mentre per quello associato verranno indicate solo in caso di variazione di tali riferimenti.

2. Studio del problema

In virtù delle specifiche fornite, le problematiche che il software dovrà affrontare sono le seguenti:

- **Registrazione utente** – Previo parsing dei dati personali (con validazione di mail e codice IBAN) ed annessa scrittura su file dedicato, ogni nuovo utente verrà registrato: nello specifico occorre controllare che l'IBAN inserito sia valido e che la mail non sia già depositata per l'accesso di un altro cliente.
- **Login utente** – Dopo una validazione da parte del software (mediante mail e password), l'utente può accedere al menu principale, in cui è possibile scegliere tra le varie opzioni. In nessun altro modo un utente deve poter accedere alle utility che il programma offre alla clientela.
- **Ordinazione merce** – In linea con le specifiche, un ordine può coinvolgere soltanto una tipologia di oggetti per volta (quindi occorrerà un filtro di selezione appositamente creato) e la scrittura potrà avvenire solo successivamente. Dovrà, inoltre, essere garantita la possibilità di poter includere in futuro una nuova categoria di merce.
- **Utility di selezione carrello** – Nel caso fosse già presente nel carrello almeno un elemento caricato, si potrà scegliere una tra le seguenti azioni:
 - aggiungere al carrello altri elementi (della stessa tipologia), mostrando di volta in volta le info legate al nuovo inserimento (prodotto, prezzo unitario, quantità, etc.);
 - visualizzare contenuto del carrello (mostrando l'insieme di oggetti caricati e la quantità desiderata);
 - confermare l'ordine impostato (successivamente a cui potrà avvenire ufficialmente la presa in carico dell'ordine);
 - cancellazione del carrello, con successivo ritorno al menu principale;
 - uscita dal menu, con la annessa cancellazione del carrello (per motivi di consistenza del database degli ordini).

- **Sezione di pagamento** – In questa sezione sarà necessario mostrare la lista degli ordini non ancora saldati (con immediata classificazione “Cliente associato/Nuovo cliente”), nonché la visualizzazione dell’IBAN attualmente abilitato al pagamento; l’utente deve anche poter scegliere quale ordine pagare, nel caso avesse impostato più ordini.
- **Calcolo del coefficiente di gradimento** – Questo parametro riguarda: l’importo totale speso attraverso il sito, il numero degli acquisti ed il numero di giorni trascorsi tra la conferma dell’ordine ed il relativo pagamento. Esso potrà essere utilizzato in futuro anche per eventuali promozioni/sconti *ad personam*.
- **Sezione di cambio IBAN** – L’utente registrato, previa validazione del nuovo codice IBAN immesso, deve poterlo aggiornare in tempo reale; questa utility, sebbene non prevista dalla specifica, risulta necessaria per garantire al cliente una misura ragionevole di flessibilità nei pagamenti.

L’elaborazione di tali problematiche, come risulta evidente in più punti, ha tenuto conto delle possibili revisioni e degli sviluppi eventualmente applicabili in futuro: in primo luogo, la scelta dell’utilizzo della tipologia di file-database di formato ‘.csv’ (*Comma Separated Values*) rende tutti i file trattati utilizzabili da fogli di calcolo elettronici (e conseguentemente da tutte le operazioni sia di gestione che di trattamento dati previsti da questi ultimi), facendo in modo che le info trattate siano utilizzabili da più apparati software.

Oltre alle preferenze progettuali riguardanti il database, si è scelto altresì di utilizzare e definire gli oggetti in maniera tale che ogni caratteristica del software risulti modulare e incrementabile.

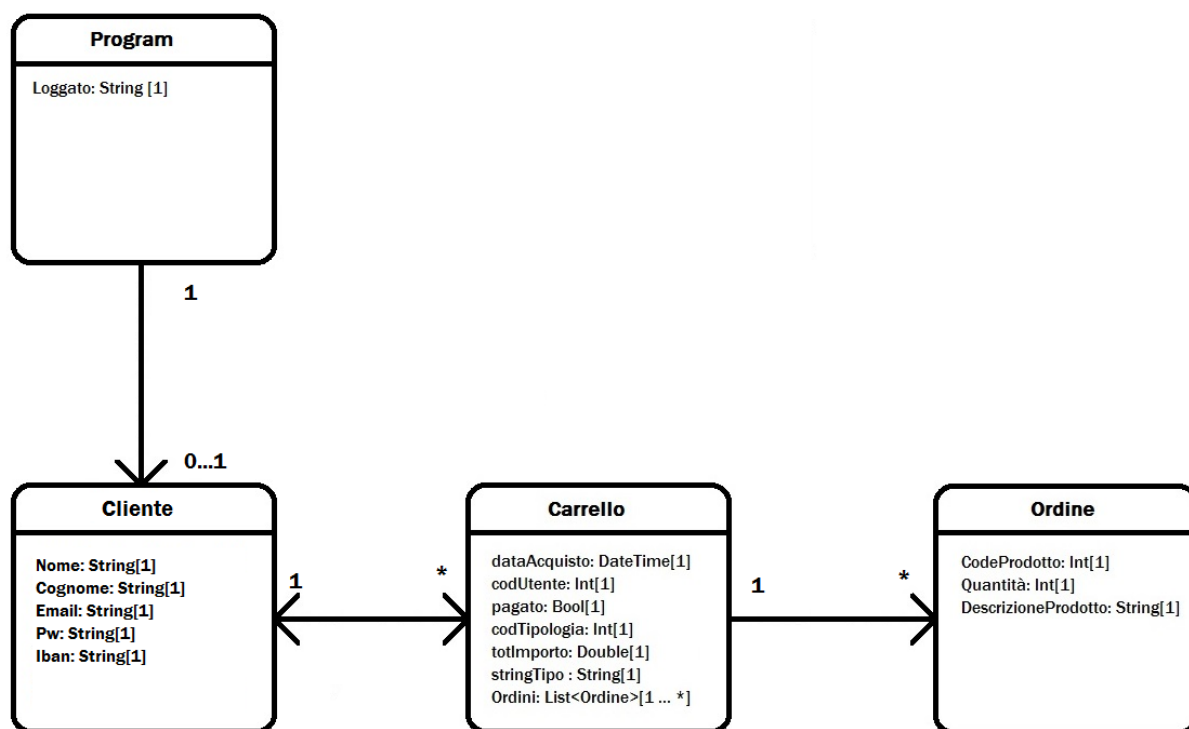
3. Scelte architetturali

Le specifiche del problema, in virtù di ciò che è venuto alla luce durante la fase di studio, hanno indirizzato lo sviluppo verso un'architettura simil *client-server*, in cui il controllo generale della struttura è destinato al programma (nella fattispecie, alla classe *Program.cs*), mentre l'utente può soltanto richiedere la fattibilità di un'operazione (sia essa su file o sull'eseguibile stesso). Questo *modus-operandi* risulta necessario affinché possano essere validate le info immesse dall'utente e si possa preservare la consistenza del database, anche in virtù del fatto che la natura dei file di memorizzazione riguarda eseguibili totalmente differenti.

Le classi partecipano orizzontalmente a tutte le fasi dell'esecuzione, per cui è pressoché impossibile una netta distinzione tra di loro a livello di contesto d'esecuzione, in quanto in determinate frazioni temporali esse potrebbero essere coinvolte in coppia o addirittura tutte: un esempio eloquente è il caso dell'aggiunta di un ordine di più prodotti su file. In questo frangente, infatti, le classi *Cliente.cs*, *Ordine.cs* e *Carrello.cs* debbono necessariamente collaborare tra di loro ed eventualmente scambiare i propri dati con uno stesso file.

4. Diagramma delle classi

Il diagramma delle classi tiene conto principalmente sia delle scelte architettoniche pianificate sia delle esigenze relative a sviluppi futuri, per cui le classi utilizzate risultano principalmente di due tipi: classi gestore (*Program.cs*), in cui i dati vengono manipolati o validati, e classi oggetto (*Cliente.cs*, *Carrello.cs* e *Ordine.cs*) mediante le quali si riesce a gestire i dati utilizzati dal programma e quelli provenienti dai file-database oppure diretti ai medesimi.



La bidirezionalità tra le classi *Cliente.cs* e *Carrello.cs* si rende necessaria al fine di garantire la navigabilità all'interno del programma e per consentire la reperibilità di dati incrociati che sarebbe, altrimenti, irrealizzabile.

Questa scelta è condizionata dal fatto che qualsiasi sviluppo futuro potrebbe richiedere un accesso da ambo le parti a scopi organizzazionali o prestazionali.

5. Documentazione per l'utilizzo

La classe *Program.cs*, come già evidenziato, gestisce tutti i metodi necessari alla validazione e gestione delle info specificate dall'utente. Tra i vari metodi che la compongono, i più importanti sono:

- **Ordina**: permette all'utente loggato di poter effettuare acquisti. Tale metodo, infatti, viene invocato soltanto dopo la validazione dell'utente;
- **Controllolban**: esegue il controllo sul codice IBAN immesso dall'utente all'atto della registrazione e nel caso in cui l'utente intenda avvalersi dell'utility di modifica IBAN di riferimento;
- **ControlloMail**: controlla se la mail di registrazione è già presente nell'archivio utente, al fine di evitare ambiguità;
- **LeggiFile**: metodo 'di servizio', esclusivamente utilizzato al fine di recuperare informazioni dai file database;
- **Menu**: mostra all'utente un menu di possibili scelte, sulla base di un elenco di opzioni disponibili in due forme (array di stringhe o lista di stringhe, in overloading);
- **AggiungiProdotto**: consente all'utente di aggiungere un prodotto ad un carrello già esistente (quindi, avente già almeno un oggetto al suo interno). Sebbene non previsto dalle specifiche, è possibile modificare tale metodo al fine di consentire acquisti su più tipologie di materiale;
- **VisualizzaCarrello**: consente all'utente di visualizzare l'attuale contenuto del carrello; questa sezione prende in considerazione tale contenuto indipendentemente dalla categoria di appartenenza dell'oggetto.

5.1 Aggiunta di prodotti

Le scelte architetturali del software consentono l'aggiunta di prodotti destinati alla vendita, seguendo la procedura sottostante: in virtù del seguente ordine semantico del file ListinoPrezzi.csv:

Codice_Tipologia; Nome_tipologia; Progressivo_Oggetto; Nome_Oggetto; Prezzo

per aggiungere altri prodotti in vendita è sufficiente:

- 1) aggiungere la tipologia, prestando attenzione a far coincidere Codice_Tip. e Nome_Tip. (es. la tipologia "1" avrà sempre "Componentistica" nel campo "Nome");
- 2) il campo "Progressivo_Oggetto" deve avere sempre il numero successivo all'ultimo campo già presente in archivio (es. se l'ultimo oggetto presente ha progressivo "12", l'oggetto che si inserisce dovrà avere progressivo "13");
- 3) per nessun motivo, potranno essere presenti all'interno dei nomi e dei codici i caratteri ";" oppure ",", e nemmeno altri caratteri speciali. Questa norma è volta a mantenere l'integrità dei database;
- 4) non è consentito lasciare righe vuote e/o campi vuoti all'interno del database dei prodotti (anche in virtù del fatto che ogni campo è vitale per il funzionamento del programma e per l'identificazione corretta dell'oggetto).

Il software, inoltre, consente l'aggiunta futura di molteplici categorie di prodotti, in seguito ad una modesta quantità di modifiche.

5.2 Gestione degli altri file

In merito ai file di gestione (tutti tranne il listino) ci si può limitare alla visualizzazione dei medesimi, in quanto il software è completamente in grado di gestire i record inseriti: l'utilizzo di tali file, dunque, può limitarsi a mero scopo di consultazione.

5.3 Esecuzione del programma

L'applicazione comprende principalmente due elementi:

- l'eseguibile (AziendaMetalmeccanica.exe) e
- la cartella di gestione (Database), contenente i file sopracitati.

Per l'esecuzione è sufficiente lanciare l'eseguibile, e si consiglia di non intervenire sui file mentre l'applicazione è in funzione.

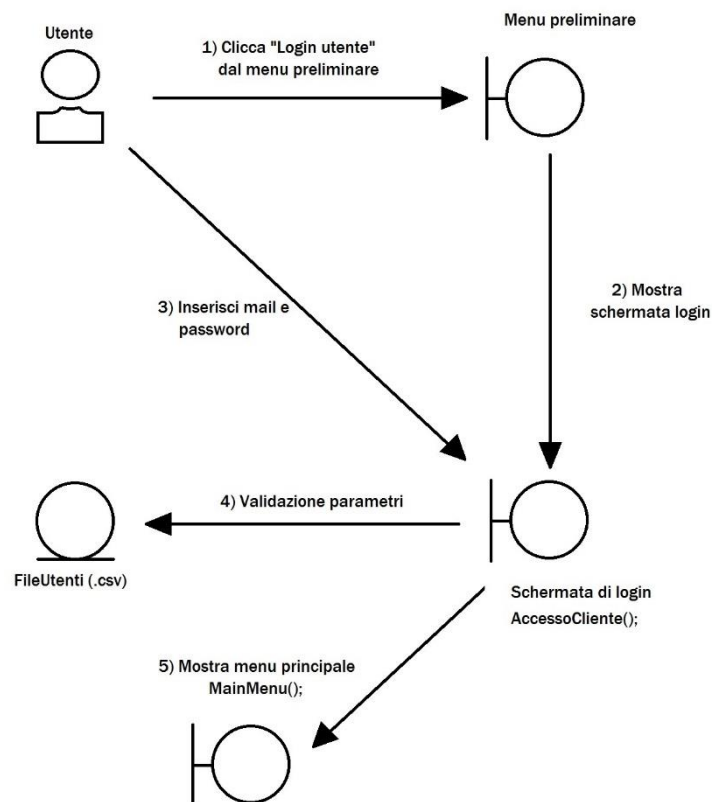
Per la modifica dei file (descritta nella pagina precedente), occorre seguire sempre le regole indicate, assicurandosi che l'eseguibile non sia stato precedentemente lanciato.

6. Use Case UML

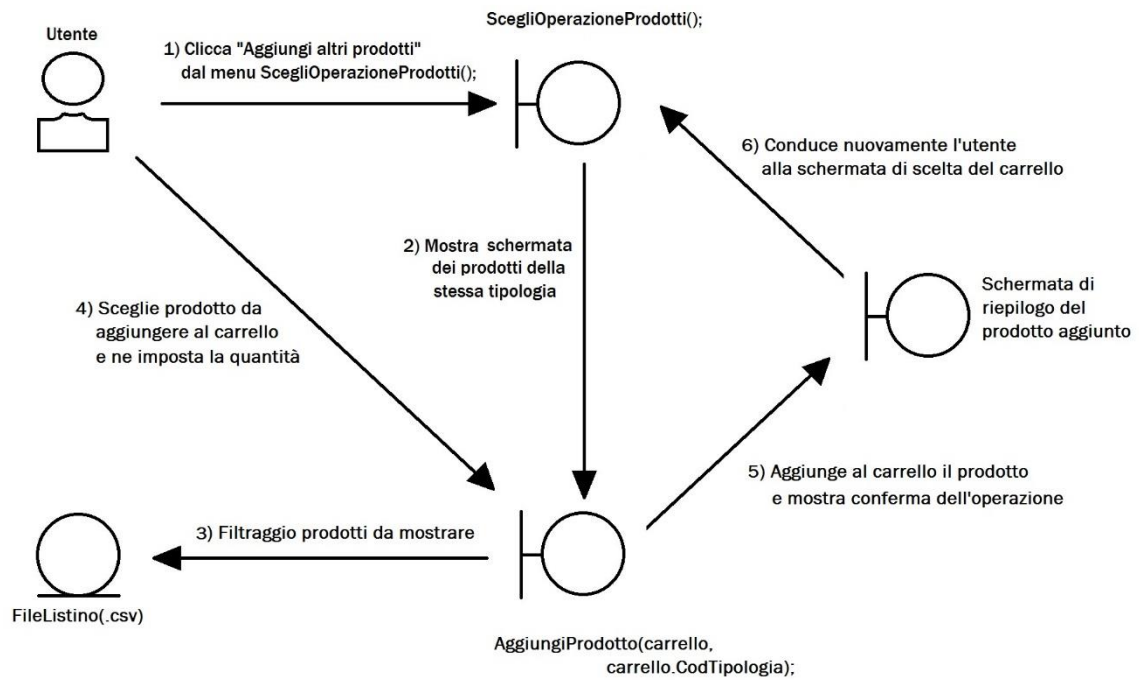
La tipologia del presente progetto rende possibile il verificarsi di una grande quantità di casi d'uso, soprattutto nell'eventualità che il software venga a trovarsi con archivi ampiamente popolati di prodotti, oppure con altre categorie di oggetti non trattati nelle specifiche; questa evenienza, tutt'altro che impossibile, ha indotto la progettazione a tenerne conto; questo aspetto, unito alla varietà di contesti possibili, ha giustificato la rappresentazione grafica dei casi d'uso in considerazione. L'analisi dei casi d'uso, dunque, si concentra principalmente su due contesti:

- a) Login di un utente già registrato
- b) Aggiunta di un prodotto ad un carrello già popolato

Caso a): Login di un utente già registrato



Caso b): Aggiunta di un prodotto ad un carrello già popolato



Indice

1. Specifiche del software	pag. 1
2. Studio del problema	pag. 2
3. Scelte architetturali	pag. 4
4. Diagramma delle classi	pag. 5
5. Documentazione per l'utilizzo	pag. 6
5.1. Aggiunta di prodotti	pag. 7
5.2. Gestione degli altri file	pag. 8
5.3. Esecuzione del programma	pag. 8
6. Use Case UML	pag. 9