

# Relazione programma Gestionale

---

## Introduzione, obiettivi e motivazioni del progetto

---

Il programma sviluppato rientra nella categoria dei software gestionali, ovvero quei programmi atti ad automatizzare alcuni processi di gestione all'interno di un'azienda.

La scelta di sviluppare un programma di questa categoria deriva dalla volontà di utilizzare gran parte delle nozioni apprese durante il corso per risolvere un problema di tutti i giorni. In particolare, si vede l'utilizzo di: formattazione di stringhe, allocazione dinamica, strutture ed enumeratori, alberi binari di ricerca (inoltre non sono molto bravo a crearmi problemi da risolvere).

Questo software gestionale è stato concepito per essere utilizzato all'interno di un negozio di piccole-medie dimensioni, pertanto alcuni controlli sull'input o azioni sono state lasciate deliberatamente poco controllate.

È possibile sintetizzare le funzionalità dell'applicativo in questione nel modo seguente:

Appena aperto, il programma funziona in modalità cliente.

È possibile per un cliente:

- Visualizzare il messaggio Novità impostato da un dipendente
- Visualizzare i prodotti in magazzino ordinandoli per ID, nome, prezzo

È possibile effettuare l'accesso come dipendente inserendo il giusto numero nella scelta del menu' iniziale, facendo questo si potrà accedere con un account dipendenti inserito precedentemente (è necessario conoscere l'id e la password).

Una volta effettuato l'accesso il programma verrà eseguito in modalità dipendente fino a che non si chiude il programma o non si effettua il logout.

Un dipendente può:

- Visualizzare e salvare il messaggio Novità
- Visualizzare con gli stessi criteri visti prima, aggiungere, modificare, eliminare i record prodotti
- Visualizzare (ordinamenti disponibili per nome cognome ID), aggiungere, modificare, eliminare i record clienti
- Visualizzare (ordinamenti disponibili per nome cognome ID), aggiungere, modificare, eliminare i record dipendenti

Se non si hanno record dipendenti disponibili per il login, verrà automaticamente introdotto il record di default per non "rimanere fuori".

Record default:

ID: 1

Password: default

---

## Architettura e progettazione dell'applicazione

---

### **Dal punto di vista utente:**

Il software si presenta strutturato a menu' con voci numerate.

Viene chiesto all'utente di effettuare la scelta attraverso l'input di un numero corrispondente alla scelta effettuata.

Arrivati all'esecuzione di funzioni diverse dalla semplice scelta dell'azione, verranno mostrate "maschere" per l'input guidato di informazioni (es. aggiunta, modifica di un record etc.).

### **Dal punto di vista strutturale:**

Il programma risolve il problema generale suddividendolo in sotto-problemi, gestiti ognuno da uno o piu' metodi.

Sono presenti metodi accessori creati ad hoc per svolgere azioni ricorrenti come il prelievo di un intero record da un determinato file, la ricerca della riga giusta in un file etc.

I file di codice sono stati divisi in base all'inerenza del contenuto (es: file.c contiene i metodi per aggiungere, eliminare, modificare record dai file + altri metodi accessori).

---

## Sviluppo e implementazione

---

Il programma e' stato sviluppato pensando di dividere l'intera esecuzione in 4 parti:

- Gestione della navigazione
- Gestione del "database"
- Strutture dati
- Ordinamenti

### **Gestione della navigazione:**

La gestione della navigazione puo' essere vista come un percorso ramificato dove ad ogni menu' viene chiesto di decidere quale strada intraprendere.

Questa parte del programma viene gestita principalmente da due metodi: uno per la gestione dell'input in modo che sia consistente col menu' visualizzato, e un altro per il setting delle varie variabili e la chiamata al metodo di visualizzazione del menu' corretto.

E' presente una variabile struttura sessione che contiene le informazioni chiave per una corretta navigazione tra cui la scelta effettuata e il tipo di menu' visualizzato o da visualizzare; inoltre contiene informazioni relative al record dipendente con cui si potrebbe aver effettuato l'accesso.

### **Gestione del “database”:**

La parte relativa al “database” e’ gestita utilizzando 3 diversi file (sono solo dei file testuali per questo ho messo le virgolette) ognuno contenente le informazioni relative ad una possibile tipologia di record. I record sono composti da campi di dimensioni fissate.

Si e’ cercato di ottimizzare l’accesso random dei file gestendo i record in modo da essere sempre in ordine di ID, in questo modo se non sono presenti “buchi” i metodi di ricerca riga o ID possono essere molto piu’ performanti.

Dato che i record dei clienti differiscono da quelli dipendenti solo per il campo password, essi sono stati gestiti come dipendenti senza password, in questo modo il codice relativo alla gestione dei record dipendenti puo’ essere utilizzato anche per record clienti (infatti si utilizza un tipo struttura persona per riferirsi a entrambi).

La scrittura su file viene effettuata utilizzando il write e non l’append in quanto non ero a conoscenza dell’effettivo funzionamento di quest’ultimo, il che ha comportato ad una gestione manuale della scrittura e riscrittura su file per emulare il funzionamento corretto dell’idea base di gestione.

Infatti, il programma per effettuare un inserimento di un record in mezzo al file andra’ a scrivere sopra alle informazioni gia’ presenti, naturalmente recuperando prima quello che verrebbe perso per poi riscriverlo in seguito. O ancora, per eliminare un record, andra’ a scrivere tutti i record presenti, su un altro file, saltando quello da eliminare.

### **Strutture dati:**

Lo sviluppo del programma prevedeva la creazione di diverse strutture ed enumeratori ad hoc per facilitare la scrittura del codice e la comprensione delle idee che stanno dietro ai metodi principali; inoltre in questo modo e’ stato possibile creare alcuni metodi utili per rendere piu’ semplice l’estrazione dei dati da file.

Le strutture presenti sono (Ulteriori dettagli sono presenti nella documentazione doxygen):

- Persona (ID, nome, cognome, password)
- Prodotto (ID, nome, prezzo, quantita’)
- Sessione (ID dipendente, password dipendente, scelta, menu)
- Tree\_Node(Prodotto, Persona, Nodo sinistro, Nodo destro)

Gli enumeratori presenti sono (Ulteriori dettagli sono presenti nella documentazione doxygen):

- campo\_record (Nome, Cognome, Id, Alfabetico, Prezzo)
- menu (Nolog, Log, Novita, Prodotti, Clienti, Dipendenti, OrdinamentoP, OrdinamentoC, OrdinamentoD)
- tipofile (Prodotto, Cliente, Dipendente)

### **Ordinamenti:**

Questa parte comprende la visualizzazione dei record di un file, ordinata in base a un campo scelto dall'utente.

E' stato utilizzato un albero binario di ricerca opportunamente popolato in modo tale che la visita inorder stampi in ordine corretto i record.

---

## **Collaudo**

---

Il programma e' stato testato emulando un possibile utilizzo reale e in alcune situazioni critiche come: assenza di file, file vuoti, input non consistente, varie situazioni di inserimento/modifica/eliminazione di vari record.

E' possibile che alcuni casi critici non siano stati testati in quanto molto improbabili per il tipo di utilizzo per cui e' pensato il programma.

E' possibile vedere possibili bug conosciuti all'interno della documentazione doxygen.

---

## **Valutazione e conclusioni**

---

Passare dall'idea iniziale alla progettazione e sviluppo del software e' stato piu' difficile di quanto previsto, ma mi reputo soddisfatto del risultato finale e di quanto interesse mi abbia suscitato il progetto nella sua interezza.

Rimangono ancora aspetti migliorabili ed implementazioni aggiuntive, per questo in fase di sviluppo sono stati creati anche alcuni metodi non utilizzati e variabili di tipi specifici per alcune idee.