

SISTEMA DI GESTIONE SARTORIA

INDICE

INDICE	1
SCOPO DEL PROGRAMMA	1
ANALISI CONTESTO	2
IPOTESI	3
ARCHITETTURA DEL SISTEMA	3
REQUISITI FUNZIONALI	4
STRUTTURE DATI	4
OPERAZIONI SUI DATI	7
FUNZIONALITÀ PRINCIPALI	8
GESTIONE DELLA PERSISTENZA	8
NORME DI SICUREZZA	9
ASPETTI DI OTTIMIZZAZIONE	9
CONSIDERAZIONI TECNICHE	9
CONCLUSIONI	10

SCOPO DEL PROGRAMMA

Il progetto ha come obiettivo la realizzazione di un sistema informatico completo per la gestione dei rotoli di tessuto all'interno di una sartoria. Il programma principale, sviluppato in linguaggio **C**, consente di registrare, monitorare e ottimizzare l'utilizzo dei materiali tessili, riducendo gli sprechi e migliorando l'efficienza operativa. A supporto, vengono integrati **strumenti web** (HTML, CSS, JavaScript) per fornire un'interfaccia moderna e interattiva. L'intero sistema è pensato per evolversi in una piattaforma modulare, capace di connettersi con servizi online (come **GitHub** e **Visual Studio** per la progettazione e lo sviluppo collaborativo).

ANALISI CONTESTO

Posizione

La sartoria si trova a **Milano**, in zona **Porta Venezia**, un quartiere centrale caratterizzato da boutique, atelier e studi di design. L'attività è inserita nel contesto urbano e lavora sia con clienti privati che con piccole aziende di moda locali. L'attività occupa circa **120 m²**.

Dimensione della sartoria

È una **sartoria artigianale di medie dimensioni**, con un laboratorio interno e un piccolo spazio showroom per i clienti. Realizza **abiti su misura**, riparazioni e collaborazioni con stilisti emergenti.

Il team è composto da **6 persone**:

- 1 titolare/sarto principale
- 2 sarti specializzati
- 1 apprendista
- 1 addetta alla clientela e appuntamenti
- 1 responsabile amministrativo

Software

Il software gestionale è **nuovo**, sviluppato per **digitalizzare i processi interni** e migliorare l'efficienza della sartoria. Prima venivano utilizzati **moduli cartacei** per appuntamenti, ordini e misure dei clienti. Il nuovo sistema permette di archiviare tutto in formato **digitale**, facilitando ricerca, aggiornamenti e condivisione dei dati.

Da cartaceo a digitale

Il passaggio è avvenuto tramite **analisi delle schede cartacee** e **interviste con il personale** (sarti, receptionist e titolare) per comprendere le esigenze reali e replicare la logica del lavoro manuale nel software. L'obiettivo è **automatizzare la gestione clienti, ordini, misure e stato di lavorazioni**, mantenendo la semplicità dell'approccio tradizionale.

IPOTESI

- **Identificazione rotoli:** Ogni rotolo ha un codice alfanumerico unico che serve a distinguerlo da tutti gli altri.
 - **Misurazioni:** Tutte le lunghezze dei tessuti sono espresse in metri.
 - **Riutilizzo dei rotoli:** Un rotolo può essere usato in più progetti diversi nel tempo. Non è "monouso".
 - **Ritagli riutilizzabili:** I pezzi di tessuto tagliati con lunghezza $\geq 0,50$ m possono essere riutilizzati.
 - **Scarti:** I pezzi più piccoli di 0,30 m sono considerati scarti, quindi non riutilizzabili.
 - **Capacità massima:** Il sistema può gestire fino a 500 rotoli contemporaneamente.
 - **Associazione prelievi:** Ogni prelievo di tessuto deve essere collegato a un progetto o cliente, identificato tramite partita IVA o Codice Fiscale.
 - **Persistenza dati:** Tutte le informazioni vengono salvate su file binario, così restano anche dopo lo spegnimento del sistema.
 - **Rotoli finiti:** Un rotolo finito rimane registrato fino a 30 giorni, con aggiornamento automatico e registrazione in log per motivi di controllo; l'operatore può comunque intervenire se necessario.
-

ARCHITETTURA DEL SISTEMA

L'architettura è **multilivello**, suddividendo le responsabilità tra componenti:

1. **Livello applicativo (C):** Contiene la logica principale per la gestione dei dati, le funzioni di calcolo, e le operazioni sui rotoli e sui prelievi di tessuto.
 2. **Livello di persistenza:** Utilizza file binari locali per il salvataggio dei dati.
 3. **Interfaccia utente web:** Realizzata in **HTML, CSS e JavaScript**, permette una consultazione e una gestione intuitiva dell'inventario anche da browser o smartphone.
 4. **Servizi di programmazione:**
 - **GitHub:** gestione del codice sorgente, versionamento e controllo delle modifiche.
 - **Visual Studio Code:** ambiente di sviluppo per coordinare programmazione, testing e debugging.
 5. **Estensioni future:** Integrazione con scanner QR o RFID per identificazione dei rotoli, e moduli predittivi per l'analisi dei consumi e un database.
-

REQUISITI FUNZIONALI

Gestione dei rotoli di tessuto:

- Inserire nuovi rotoli in magazzino.
- Modificare i dati dei rotoli esistenti.
- Eliminare rotoli non più presenti o scaduti.
- Visualizzare l'elenco completo dei rotoli.

Gestione dei prelievi:

- Registrare i prelievi di tessuto per ciascun progetto/cliente.
- Aggiornare automaticamente la quantità residua del rotolo.

Gestione ritagli e scarti:

- Classificare i ritagli in riutilizzabili ($\geq 0,50$ m) e scarti ($< 0,30$ m).
- Monitorare la disponibilità dei ritagli riutilizzabili.

Ricerca e filtraggio:

- Cercare rotoli in base a tipo, colore, lotto o lunghezza residua.

Ottimizzazione dei tagli:

- Calcolare suggerimenti di taglio per ridurre gli sprechi di tessuto.

Storico delle operazioni:

- Tenere traccia di tutti i prelievi, modifiche e scarti.
- Fornire report storici su prodotti e azioni effettuate.

Interfaccia utente:

- Consentire l'interazione tramite interfaccia web chiara e intuitiva.

STRUTTURE DATI

Nel linguaggio C, i dati vengono gestiti tramite **strutture statiche**, progettate per rappresentare in modo efficiente i rotoli, i prelievi e i ritagli. Possibile esempio:

struttura Rotolo:

- **id** (char[50]) = Identificativo univoco del rotolo di tessuto.
- **tipo** (char[50]) = Descrive la tipologia del tessuto (es. cotone, lino, seta, ecc.).

- **colore** (char[50]) = Indica il colore principale del tessuto.
- **fantasia** (char[50]) = Specifica motivo o disegno (es. righe, pois, tinta unita).
- **lunghezza_totale** (float) = Rappresenta la lunghezza totale del rotolo in metri al momento dell'acquisto.
- **residuo** (float) = Indica la lunghezza ancora disponibile dopo i vari prelievi.
- **costo_metro** (float) = Prezzo del tessuto per metro lineare.
- **fornitore** (struct Fornitore) = Struttura che contiene i dati del fornitore (Partita IVA, nome, indirizzo, contatti, ecc.).
- **lotto** (char[50]) = Codice identificativo del lotto, generato automaticamente in base al fornitore.
- **data_acquisto** (struct data) = Struttura che memorizza la data di acquisto (giorno, mese, anno).
- **stato** (char) = Indica lo stato del rotolo (es. NUOVO, ESAURITO, ecc.).
- **noteAggiuntive** (char[100]) = Campo testuale per eventuali annotazioni.

Struttura Data:

- **giorno** (int) = Rappresenta il giorno interessato.
- **mese** (int) = Indica il mese, espresso come numero da 1 a 12.
- **anno** (int) = Specifica l'anno di riferimento per la data.

Struttura Fornitore:

- **nome** (char[50]) = Nome commerciale o ragione sociale del fornitore.
- **partita_iva** (char[20]) = Codice identificativo fiscale dell'azienda o artigiano.
- **indirizzo** (char[100]) = Via, numero civico e città della sede del fornitore.
- **telefono**(char[20]) = Numero di contatto per ordini o comunicazioni.
- **email**(char[50]) = Contatto elettronico ufficiale del fornitore.

Struttura Prelievo:

- **id** (char[50]) = Codice identificativo univoco del prelievo
- **id_rotolo** (char[50]) = Collegamento al rotolo da cui è stato prelevato il tessuto.
- **metraggio_prelevato** (float) = Lunghezza del tessuto prelevata (metri)
- **data** (struct data) = Contiene giorno, mese e anno del prelievo.
- **operatore** (char[50]) = Nome dell'operatore che ha eseguito il prelievo.

Struttura Ritaglio:

- **idRitaglio** (char[50]) = Codice identificativo del ritaglio, generato automaticamente per tracciarlo nel sistema.
- **id_rotolo** (char[50]) = Riferimento al rotolo da cui il ritaglio è stato ottenuto.
- **lunghezza** (float) = Misura del ritaglio (in metri), utile per eventuali riutilizzi o vendite di scarti.
- **dataCreazione**(struct data) = Contiene giorno, mese e anno in cui il ritaglio è stato registrato.

Struttura Progetti:

- **id**(char[50]) = Codice identificativo univoco del progetto, utile per la tracciabilità.
- **idCliente**(char[50]) = Codice fiscale del cliente per collegare il progetto al destinatario.
- **tipo_capo** (char[50]) = Specifica il tipo di capo realizzato (camicia, giacca, pantalone, ecc.).
- **idRotolo** (char[50]) = Codice del rotolo principale da cui è stato prelevato il tessuto.
- **tessuto_usato**(char[50]) = Descrizione del tessuto impiegato nel progetto (tipo, colore, fantasia).
- **dataConsegna**(struct data) = Giorno, mese e anno della consegna prevista o effettiva del capo.

Struttura Magazzino:

- **data_controllo** (struct data) = Indica la data dell'ultimo inventario o controllo del magazzino.
- **rotoli** (struct rotolo[200]) = Array contenente fino a 200 rotoli presenti nel magazzino, ognuno con tutte le informazioni del tessuto.
- **valoreTotale** (float) = Valore monetario totale dei tessuti in magazzino.
- **metraggioTotale** (float) = Somma dei metri disponibili di tutti i rotoli.
- **numeroRotoli** (int) = Numero effettivo di rotoli attualmente presenti.
- **priorita_utilizzo** (int) = Indica la priorità con cui utilizzare i tessuti:
 1. prima (da utilizzare subito)
 2. normale
 3. scorta (da usare solo quando necessario)

Queste strutture sono progettate per essere compatte, facilmente utilizzabili e salvabili in un file binario, o un possibile database.

OPERAZIONI SUI DATI

Le principali operazioni implementate nel programma comprendono:

Inserimento nuovo rotolo:

1. Acquisire tutti i dati del rotolo da tastiera (codice, tipo, colore, fornitore, lunghezza totale, eventuali note).
2. Verificare che il codice rotolo sia unico.
3. Inizializzare `lunghezzaDisponibile` = `lunghezzaTotale`.
4. Impostare `statoRotolo` = NUOVO.
5. Salvare il rotolo nel database (array o file binario).

Ricerca rotoli:

Permettere ricerche filtrate secondo:

- Codice rotolo
- Tipo tessuto
- Colore
- Fornitore
- Stato del rotolo (es. NUOVO, IN USO, ESAURITO)
- Disponibilità minima (es. tutti i rotoli con `lunghezzaDisponibile` > 5 m)

Modifica rotolo:

- Consentire modifica di **note, stato e altri campi non critici**.
- **Non permettere la modifica diretta delle lunghezze:** l'aggiornamento avviene solo tramite registrazione dei prelievi.

Eliminazione logica rotolo:

- Non cancellare fisicamente il rotolo.
- Impostare `statoRotolo` = ESAURITO quando `lunghezzaDisponibile` ≤ 0.
- Questo permette di mantenere lo storico dei rotoli anche dopo che sono finiti.

FUNZIONALITÀ PRINCIPALI

- **Ottimizzazione del taglio:** Implementazione di un algoritmo che calcola il modo migliore per tagliare i tessuti, riducendo al minimo gli scarti.
 - **Gestione ritagli:** Sistema che registra i ritagli prodotti e li propone per futuri utilizzi compatibili.
 - **Tracciabilità digitale:** Ogni rotolo può essere identificato con un codice QR, che collega direttamente ai dati del materiale nel database.
 - **Reportistica automatica:** Generazione di riepiloghi statistici per consumi, giacenze, fornitori e utilizzo settimanale.
 - **Interfaccia web interattiva:** Una dashboard moderna consente di visualizzare l'inventario, simulare i tagli e monitorare le operazioni in tempo reale.
-

GESTIONE DELLA PERSISTENZA

Il sistema gestisce le informazioni tramite file binari locali, garantendo velocità e persistenza:

- **rotoli.dat** → archivio principale dei rotoli.
- **prelievi.dat** → registro dei tagli e prelievi.
- **inventario.dat** → stato dei tessuti e controlli periodici di magazzino.

Ogni file viene aperto in modalità binaria e aggiornato automaticamente a ogni operazione di inserimento, modifica o eliminazione.

Questi file garantiscono:

- persistenza automatica tra le sessioni;
 - integrità tramite meccanismo di salvataggio e caricamento;
 - **backup automatici** in file binari;
 - **sincronizzazione con repository GitHub** per mantenere versioni aggiornate e verificabili del codice e della struttura dati.
-

NORME DI SICUREZZA

- **Validazione input:** verifica di ogni dato inserito per evitare errori.
 - **Backup dati:** conservazione sicura delle copie dei dati in archivi esterni.
 - **log azioni:** registrazione di tutte le operazioni critiche con timestamp e nome utente.
-

ASPETTI DI OTTIMIZZAZIONE

- **Efficienza della memoria:** utilizzo di tipi di dato compatti e allocazioni controllate.
 - **Prestazioni:** utilizzo di file binari per comodità e leggerezza.
 - **Ottimizzazione degli algoritmi:** uso di tecniche di programmazione per l'ottimizzazione del taglio.
 - **Scalabilità:** possibilità di passare a un database in caso di utilizzo in un contesto reale.
-

CONSIDERAZIONI TECNICHE

- **Linguaggi utilizzati:**
 - C per la logica principale.
 - HTML, CSS, JS per l'interfaccia web.
 - Eventuale utilizzo di un database
 - **Strumenti e servizi:**
 - Visual Studio / VS Code come ambiente di sviluppo.
 - GitHub per le versioni, collaborazione e test automatici.
-

CONCLUSIONI

Il progetto rappresenta un sistema completo e moderno per la **digitalizzazione del processo gestionale di una sartoria**. Le funzionalità rendono questo progetto non solo utile a livello scolastico, ma potenzialmente applicabile anche in contesti professionali reali.

La gestione del sistema permetterà in futuro di integrare ulteriori estensioni, come l'analisi predittiva dei consumi e la connessione con macchinari tessili automatizzati, rendendo la sartoria totalmente digitale.