

---

# **Sistema di Gestione Inventario e Magazzino**

**per Piccole Attività**

---

**Gruppo:**

Corsini Cristian  
Feola Simone  
Spitaleri Daniela

# Indice

<b>1 Obiettivo del Progetto</b>	<b>2</b>
<b>2 Documentazione di Processo</b>	<b>2</b>
2.1 Strumenti Utilizzati . . . . .	3
2.2 Organizzazione degli Sprint . . . . .	3
<b>3 Documentazione Tecnica</b>	<b>3</b>
3.1 Architettura MVC . . . . .	3
3.2 Struttura dei File (Source Code) . . . . .	4
3.3 Schema Database . . . . .	4
<b>4 Manuale Utente</b>	<b>5</b>
4.1 Primo Accesso . . . . .	5
4.2 Dashboard Principale . . . . .	5
4.3 Carico e Scarico Merce . . . . .	6
4.4 Visualizzazione Statistiche . . . . .	6

# Guida alla Prima Configurazione

Al primo avvio dell'applicazione su un nuovo computer, è necessario configurare la connessione al database MySQL affinché il software possa leggere e salvare i dati correttamente.

Seguire questi passaggi:

1. Avviare l'applicazione.
2. Nella barra dei menu in alto, cliccare su **Main > Impostazioni**.
3. Si aprirà la finestra di dialogo "**DAO Settings**".
4. Compilare i campi con i parametri del proprio server MySQL locale o remoto:
  - **Host:** Indirizzo IP del server (es. `localhost` o `127.0.0.1`).
  - **Username:** Nome utente del database.
  - **Password:** Password dell'utente database.
  - **Schema:** Nome del database creato (es. `magazzino`).
5. Cliccare su **OK** per salvare.

*Nota: Se la connessione ha successo, l'applicazione caricherà immediatamente i dati nella schermata principale. In caso di errore, verificare che il servizio MySQL sia attivo e che le credenziali siano corrette.*

## 1 Obiettivo del Progetto

L'obiettivo del progetto è fornire alle piccole attività commerciali uno strumento semplice per gestire il proprio magazzino, eliminando la necessità di utilizzare registri cartacei.

Il software è stato progettato per risolvere i problemi tipici della gestione manuale, offrendo:

- **Ordine:** Un catalogo digitale sempre pulito e aggiornato.
- **Automazione:** Le quantità si aggiornano da sole dopo ogni carico o scarico, evitando errori di calcolo.
- **Controllo:** Il sistema avvisa quando un prodotto sta per finire, evitando di rimanere senza scorte.

## 2 Documentazione di Processo

Il progetto è stato sviluppato adottando la metodologia agile **SCRUM**. Questo approccio ha permesso di rilasciare incrementi funzionali del software alla fine di ogni iterazione (Sprint).

## 2.1 Strumenti Utilizzati

- **Taiga:** Utilizzato per la gestione del Product Backlog, la definizione delle User Stories e il tracciamento dei task.
- **GitHub:** Utilizzato per il versionamento del codice sorgente e la condivisione del lavoro tra i membri del team.

## 2.2 Organizzazione degli Sprint

Lo sviluppo è stato suddiviso in 4 Sprint principali:

### Sprint 1: Gestione Catalogo

*Obiettivo:* Creazione dell'infrastruttura base e gestione anagrafica prodotti.

- Setup architettura MVC e Database MySQL.
- Implementazione User Story: "Visualizzazione lista prodotti".
- Implementazione User Story: "Inserimento nuovo prodotto".

### Sprint 2: Gestione Movimenti

*Obiettivo:* Gestione dinamica delle giacenze.

- Implementazione logica di Carico e Scarico.
- Funzionalità di Modifica e Cancellazione prodotti.
- Gestione transazioni SQL per garantire la coerenza dei dati.

### Sprint 3: Dashboard e Reporting

*Obiettivo:* Strumenti di monitoraggio.

- Implementazione alert visivi per prodotti sotto soglia.
- Creazione filtri di ricerca.
- Esportazione dati in formato CSV.

### Sprint 4: Analisi e Sicurezza

*Obiettivo:* Rifinitura e protezione.

- Integrazione grafici statistici (JavaFX Charts).
- Implementazione sistema di Login (Autenticazione).

## 3 Documentazione Tecnica

### 3.1 Architettura MVC

L'applicazione segue il pattern **Model-View-Controller** per separare la logica di business dall'interfaccia grafica.

- **Model:** Contiene le classi `Prodotti.java`, `Movimenti.java` e il layer di accesso ai dati (DAO).
- **View:** Definita tramite file FXML, si occupa solo della presentazione grafica.

- **Controller:** Gestisce gli eventi utente (click, input) e aggiorna il Model.

## 3.2 Struttura dei File (Source Code)

Il codice sorgente, situato nel package `it.unicas.project.template.address`, è organizzato in package funzionali:

- **model:** Contiene i bean (`Prodotti`, `Movimenti`) e il sotto-package `dao` per la gestione delle query SQL e della connessione al database.
- **view:** Contiene l'interfaccia grafica (file `.fxml`) e i relativi Controller (es. `ProdottiOverviewController`) che gestiscono la logica di presentazione.
- **util:** Contiene classi di supporto trasversali, come `Esportazione.java` per la generazione dei report CSV.
- **test:** Contiene le suite di test JUnit (`ProdottiTest`, `MovimentiTest`) per la validazione del codice.

## 3.3 Schema Database

Il sistema utilizza un database MySQL relazionale composto da due tabelle principali collegate da relazione 1:N.

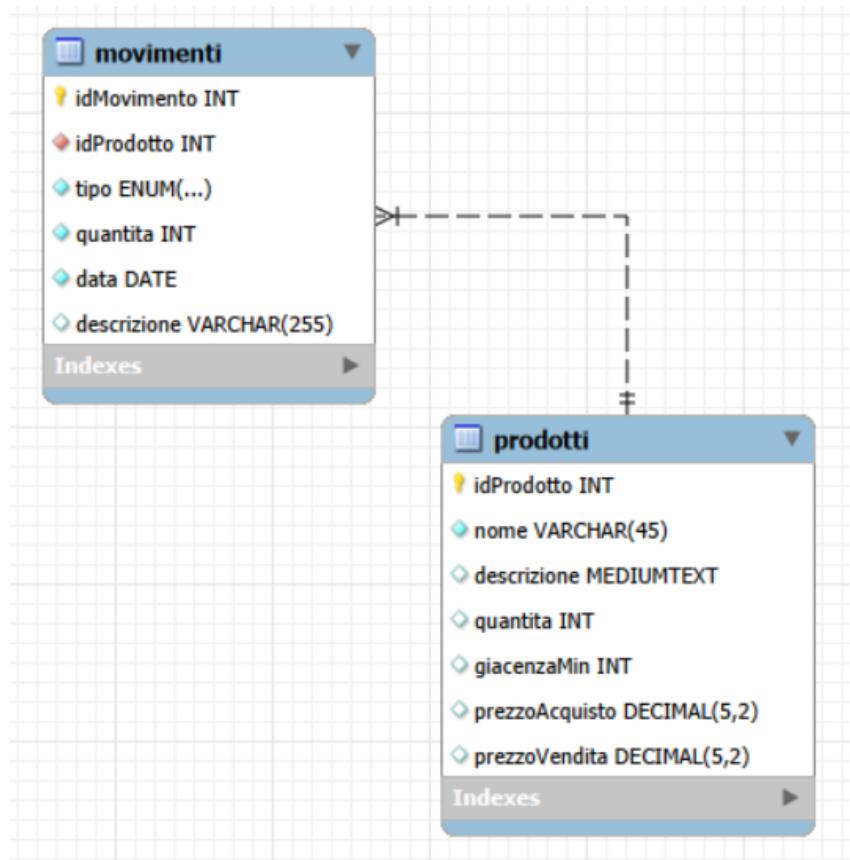


Figura 1: Schema E-R del Database: Relazione tra Prodotti e Movimenti

## 4 Manuale Utente

Questa sezione guida l'utente attraverso le funzionalità principali dell'applicazione.

### 4.1 Primo Accesso

All'apertura del programma viene mostrata la finestra di autenticazione. Inserire la password fornita dall'amministratore per sbloccare il sistema.

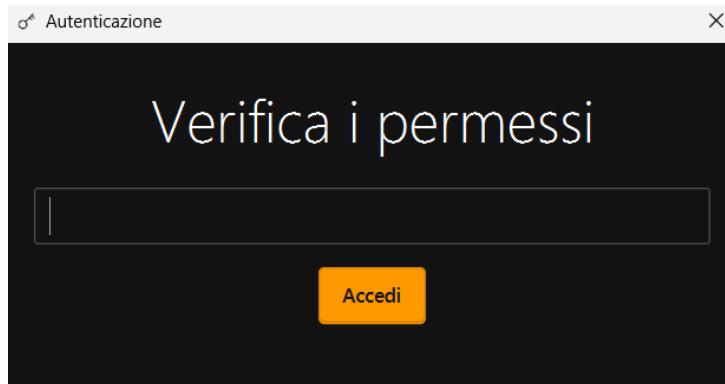


Figura 2: Schermata di Autenticazione

### 4.2 Dashboard Principale

La schermata principale è organizzata in quattro pannelli funzionali per un monitoraggio completo:

- **Elenco Prodotti (in alto a sinistra):** Tabella navigabile dell'inventario. Sotto di essa sono posizionati i pulsanti "**Ricerca**" (per filtrare i prodotti) e "**Aggiungi**" (per inserire nuove referenze).
- **Dettagli Prodotto (in alto a destra):** Mostra le specifiche dell'elemento selezionato nella lista. Qui si trovano i comandi operativi: "**Carico/Scarico**", "**Modifica**" ed "**Elimina**". Se la quantità è critica, il numero appare in rosso accompagnato dall'etichetta "**Da riordinare**" altrimenti apparirà "**Disponibile**".
- **Criticità (in basso a sinistra):** Sezione dedicata ai prodotti sotto soglia, che evidenzia immediatamente cosa manca e la quantità necessaria.
- **Ultimi Movimenti (in basso a destra):** Storico cronologico delle operazioni recenti effettuate nel magazzino.

The screenshot shows the 'Gestione magazzino' application interface. At the top, there's a navigation bar with links: Main, Statistiche, Esporta, Aiuto. Below it, the main area is divided into several sections:

- Elenco Prodotti**: A table listing products with columns for Name and Description. Examples include "Notebook HP ProBook 450 G9", "Mouse Wireless Logitech M185", and "Monitor Samsung 24" LED IPS".
- Dettagli Prodotto**: A detailed view of a selected product, "Mouse Wireless Logitech M185", showing its description as "Mouse ottico senza fili grigio/nero. Include ricevitore nano USB "Plug & Play". Batteria a ...", quantity (2), minimum stock (5), purchase price (8.5), and selling price (18.9).
- Criticità**: A table showing the criticality of products, with columns for Product and Shortage. An example row shows "Mouse Wireless Logitech M185" with a shortage of 3.
- Ultimi Movimenti**: A table showing recent movements with columns for Date, Product, Type, and Quantity. The data includes various entries for mice, notebooks, and monitors from different dates like 2025-12-13 and 2025-11-28.

Figura 3: Schermata Principale

### 4.3 Carico e Scarico Merce

Per registrare un movimento:

1. Selezionare un prodotto dalla tabella.
2. Cliccare su "Carico" (per aggiungere) o "Scarico" (per rimuovere).
3. Inserire la quantità desiderata nella finestra di dialogo.
4. *Nota:* Il sistema impedirà lo scarico se la quantità inserita supera la giacenza disponibile.

### 4.4 Visualizzazione Statistiche

Cliccando sul pulsante "Statistiche", si apre una vista grafica che mostra l'andamento mensile dei movimenti e i prodotti più popolari, utile per prendere decisioni di riordino.

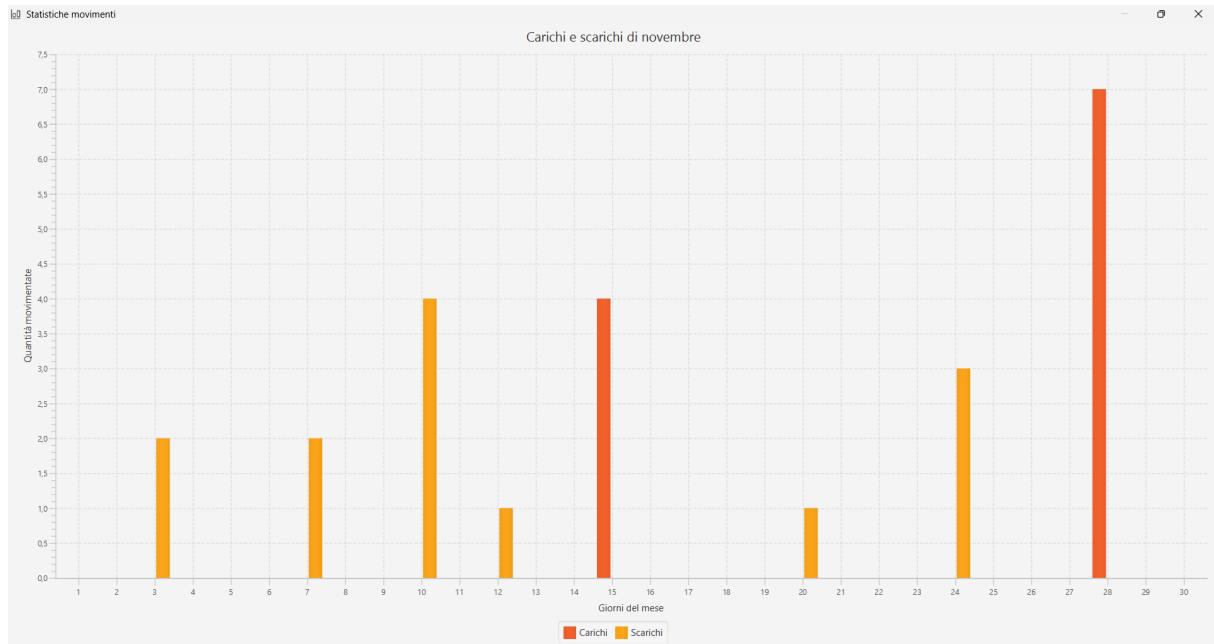


Figura 4: Carichi e scarichi dell'ultimo mese

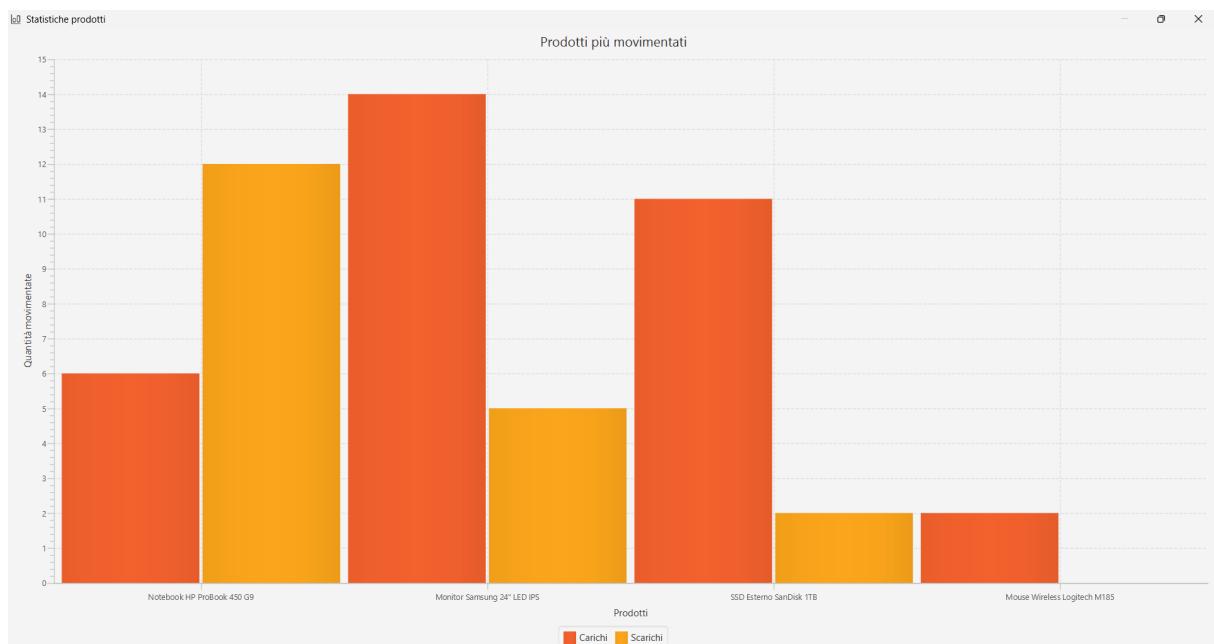


Figura 5: Prodotti più movimentati