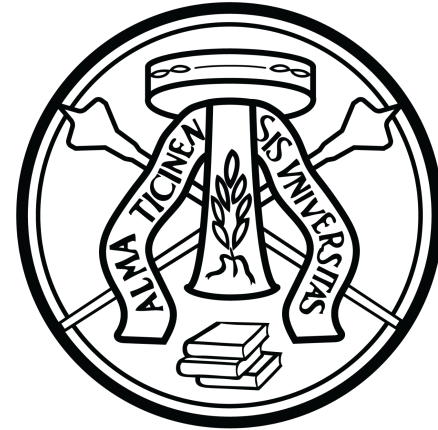


2024-2025 Programmazione ad Oggetti ed Ingegneria del Software

PROGETTO-G25

Alessandro D'Elia,
Annamaria Anzelmi,
Alice Marano,
Davide Re



UNIVERSITÀ
DI PAVIA

Introduzione

Il nostro progetto nasce dall'esigenza di semplificare l'interazione con distributori automatici, garantendo pagamenti rapidi, gestione dinamica delle scorte e un'esperienza utente semplice ed intuitiva. L'obiettivo è fornire una soluzione che risponda alle esigenze del mercato, dove la velocità e la sicurezza sono fondamentali, oltre che a quelle degli utenti, con una totale digitalizzazione del sistema di pagamento e dell'esperienza d'uso.



"BiteByte": La Rivoluzione dei Distributori Automatici".

Unified Process (UP)

UP è un processo software che incarna il modello incrementale e iterativo, perfetto ibrido del modello a cascata e di quello evoluzionistico. Esso ci ha permesso di suddividere il progetto in fasi, gestire i rischi sin dalle prime iterazioni e garantire rilasci funzionali in tempi brevi.

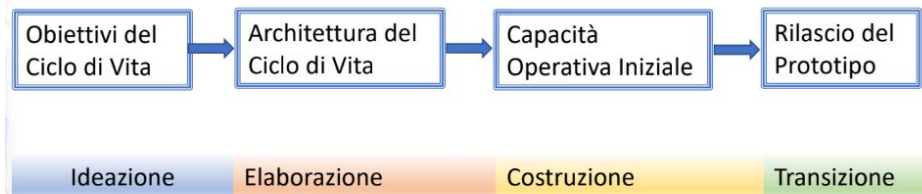
Questo approccio ha favorito una comunicazione fluida e un continuo miglioramento basato sui feedback ricevuti, oltre che un'importante versatilità di fronte a modifiche dei requisiti di sistema.

Utilizzare UP significa poter rilasciare versioni funzionanti del sistema ad ogni ciclo, verificando costantemente la qualità e adattandosi alle variazioni dei requisiti.

Iterazioni brevi: Permettono il testing costante

Mitigazione dei rischi: I potenziali problemi vengono individuati e risolti presto

Feedback continuo: Miglioramenti basati sull'esperienza degli utenti e degli stakeholder



fasi:

- Requisiti
- Analisi
- Progettazione
- Implementazione
- Test

Fase di Ideazione:

In questa fase iniziale abbiamo definito la visione del progetto, individuato gli obiettivi principali e raccolto i requisiti di base. È stata un'unica iterazione in cui abbiamo:

- Definito l'idea di simulare distributori automatici in ambiente virtuale.
- Raccolto requisiti preliminari sia funzionali (registrazione, autenticazione, visualizzazione dei prodotti, filtri, acquisto, ecc.) che non funzionali (affidabilità, sicurezza, scalabilità).
- Redatto il documento di visione.

Fase di Elaborazione:

In questa fase abbiamo approfondito e affinato i requisiti e l'architettura del sistema.

- **Iterazione 1:** Analisi dei casi d'uso, identificazione dei rischi principali, definizione preliminare dell'architettura (l'adozione di MVC, DAO, l'uso dei pattern Strategy, Composite, Factory, Facade, Singleton–) e scelta degli strumenti tecnologici (JavaFX, JDBC, ecc.).
- **Iterazione 2:** Revisione e dettaglio dei requisiti, aggiornamento dei diagrammi UML (casi d'uso, classi, sequenze) e definizione più precisa della logica di business.

Fase di Costruzione:

- **iterazione 1:** La prima iterazione ha riguardato lo sviluppo delle funzionalità base e la creazione della struttura principale del sistema.
- **Iterazione 2:** Ogni membro del team ha lavorato su casi d'uso specifici, ognuno dei quali ha portato a incrementi funzionali indipendenti.

Fase di Transizione:

- **iterazione 1:** Nella prima iterazione abbiamo effettuato il merge dell'intero sistema per verificare che tutti i componenti funzionassero insieme e che le interfacce tra i vari moduli fossero correttamente integrate.
- **iterazione 2:** Nella seconda iterazione ci siamo concentrati sull'esecuzione di test approfonditi (di integrazione, accettazione e collaudo) per individuare eventuali bug residui e garantire che il prodotto rispondesse ai requisiti richiesti.

Requisiti

Il progetto si articola su due livelli di requisiti:

Requisiti Funzionali:

Il sistema deve gestire registrazioni, autenticazioni, profili utente, portafoglio virtuale, ricerca, filtraggio, acquisti, gestione e rifornimento prodotti e notifiche.

Requisiti Non Funzionali:

È fondamentale garantire affidabilità, scalabilità, sicurezza e rapidità nelle operazioni, oltre al monitoraggio continuo degli eventi critici.

Casi d'Uso

Nella fase di analisi, abbiamo mappato ogni possibile interazione:

I principali attori sono il **Cliente** e l'**Amministratore**.

Casi d'uso significativi:

UC1. Registrazione e Accesso

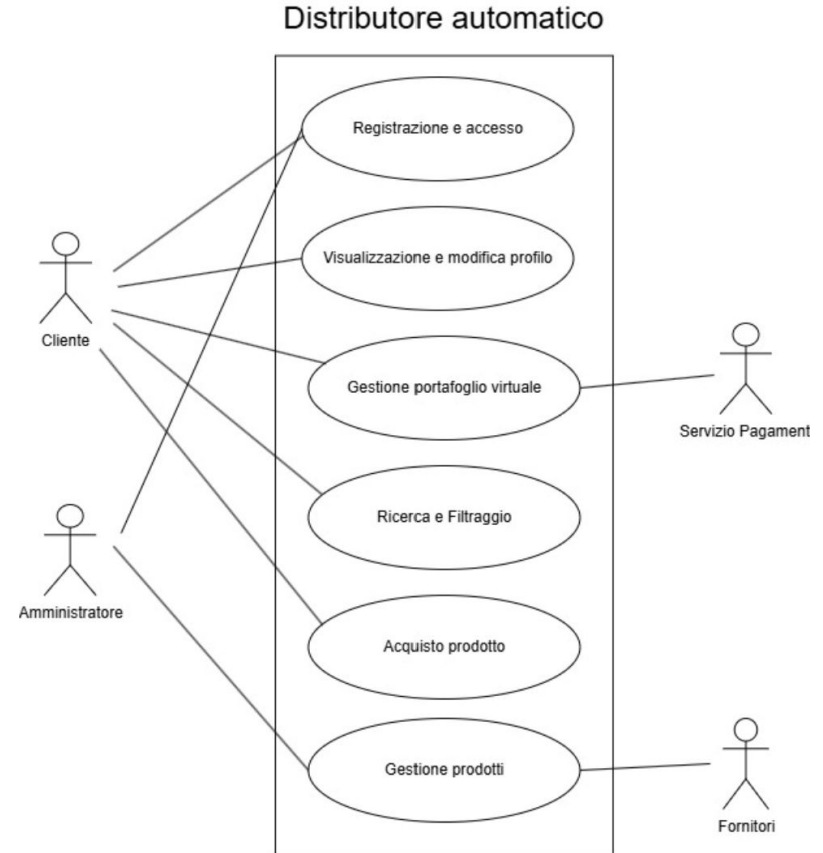
UC2. Modifica profilo cliente e Gestione portafoglio virtuale

UC3. Ricerca e Filtraggio

UC4. Acquisto prodotto

UC5. Gestione prodotti

I casi d'uso distinti per scenario sono definiti in dettaglio nel documento "Casi d'Uso.pdf".



INTERFACCE GRAFICHE

Abbiamo utilizzato il linguaggio di programmazione Java insieme al framework JavaFX (e a piccoli inserti in **CSS**) per sviluppare un'interfaccia utente grafica intuitiva e coinvolgente.

JavaFX è un framework per lo sviluppo di applicazioni desktop (e anche mobile) in Java, che permette di creare interfacce utente ricche e interattive. **Scene Builder**, invece, è uno strumento visuale che aiuta a progettare graficamente queste interfacce senza dover scrivere manualmente il codice FXML.


CSS, acronimo di "Cascading Style Sheets", è un linguaggio di stile utilizzato per definire l'aspetto e il formato dei documenti scritti in HTML, XML ecc...


Nelle slide successive forniamo piccoli esempi di interfacce in modo da dare un'idea sull'aspetto finale del sistema, sia lato Cliente che lato Amministratore

FUNZIONALITÀ

Login

Registrazione cliente






Benvenuto!

Username

Password


Log in!

Sei un nuovo cliente? **Registrati!**



Dati principali associati all'account

Data di nascita



Nome utente **Codice Fiscale**


Indirizzo Email

Password **Conferma Password**

Torna al login **Registrati!**

FUNZIONALITÀ

Visualizzazione e modifica profilo cliente connesso
Gestione portafoglio Virtuale



Profilo Cliente

Nome:

Cognome:

Nome Utente:

Data di nascita:

Email:

Codice Fiscale:

Password:

Modifica

Log out

Portafoglio

Storico Ordini

Connetti a un distributore

Modifica del Profilo Cliente

Torna al profilo cliente

Nome:

Cognome:

Nome Utente:

Password:

Salva e riconnettiti!



Portafoglio Virtuale

ID portafoglio:

Saldo:

Tipologia Pagamento:

Ricarica 5€

Torna a profilo cliente

FUNZIONALITÀ

Scelta tipologia di pagamento
Inserimento dati Bancomat

Tipologia Pagamento



Bancomat



PayPal

Bancomat

Numero Carta

Titolare

Data Scadenza

CVV

Circuito

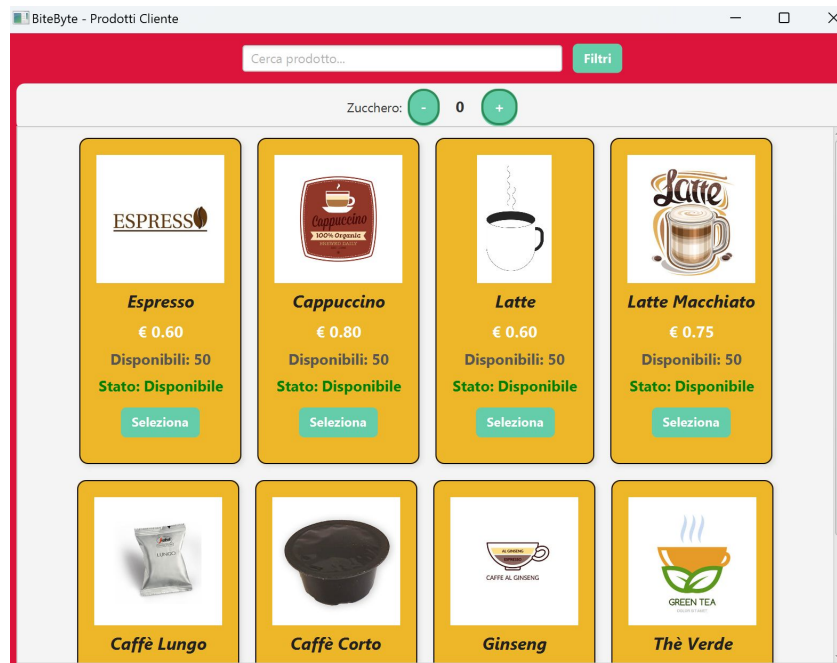
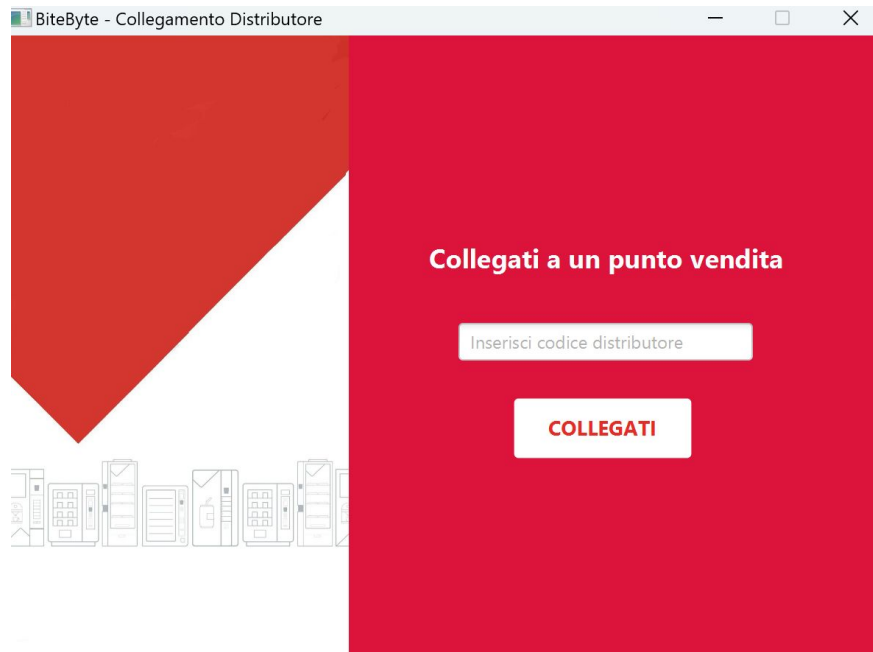


Salva

FUNZIONALITÀ

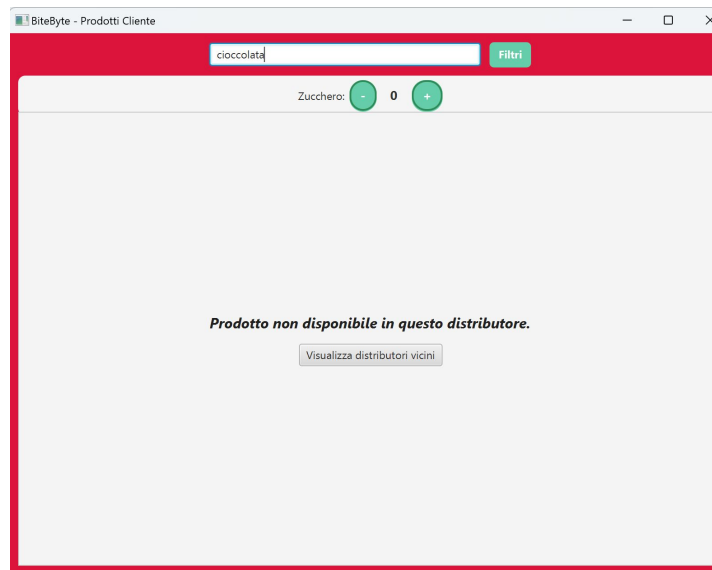
Connessione al distributore

Visualizzazione prodotti associati al distributore



FUNZIONALITÀ

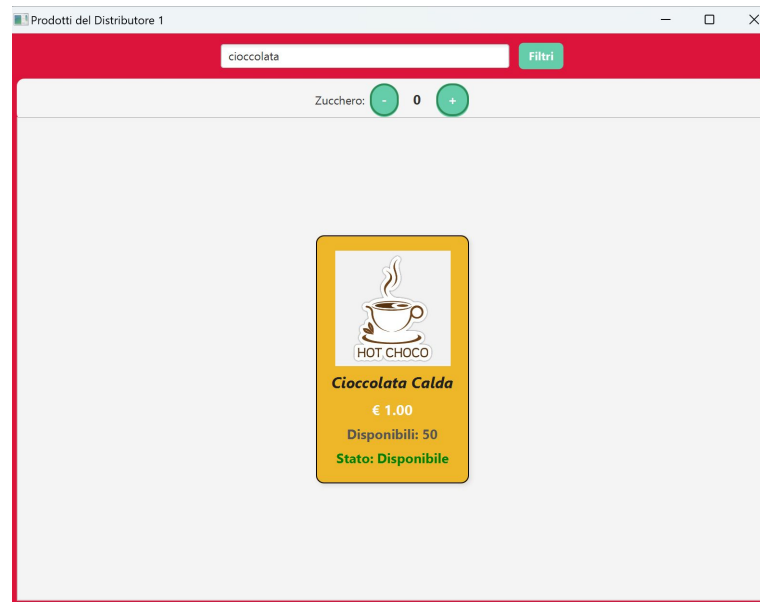
Ricerca e filtraggio dei prodotti



FUNZIONALITÀ

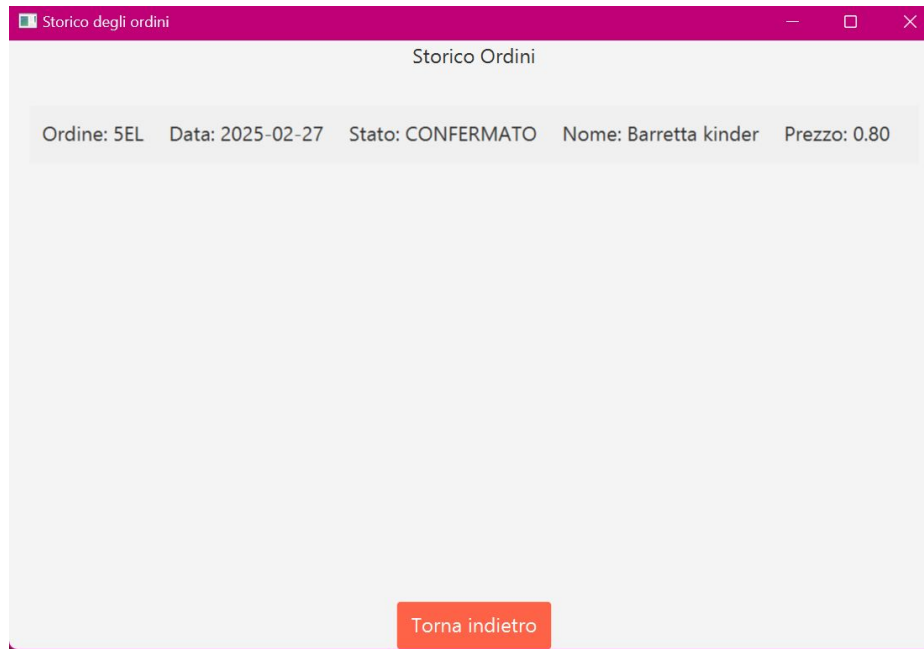
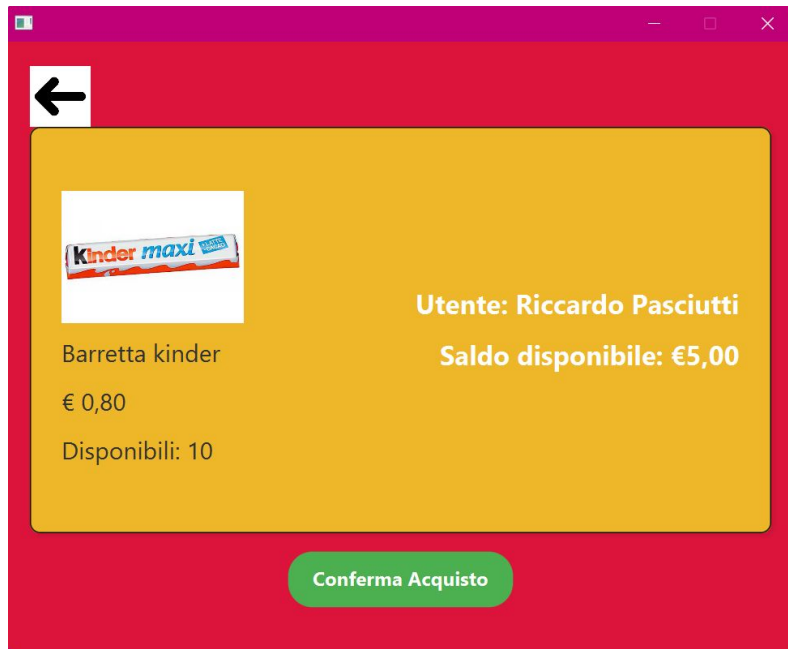
suggerire all'utente un elenco di distributori vicini da cui poter acquistare il prodotto non presente nel distributore corrente

Distributori Alternativi					
Nome	Indirizzo	Distanza (km)	Vai	Visualizza prodotto	
1	Corso Cavour 15	2.21 km	Vai	Visualizza prodotto	
5	Corso Strada Nuova 106	2.32 km	Vai	Visualizza prodotto	



FUNZIONALITÀ

Acquisto prodotto e Storico ordini



FUNZIONALITÀ

Schermata principale amministratore

Modifica prezzo e sostituzione

Gestione Prodotti

Fonzie

€ 1,00

Disponibili: 10

Stato: Disponibile

Barretta kinder

€ 0,80

Disponibili: 10

Stato: Disponibile

Patatine Classiche

€ 1,00

Disponibili: 10

Stato: Disponibile

Popcorn Salati

€ 1,20

Disponibili: 10

Stato: Disponibile

Arachidi Tostate

€ 0,60

Disponibili: 10

Stato: Disponibile

Crackers Salati

€ 0,80

Disponibili: 10

Stato: Disponibile

Nachos Piccanti

€ 1,00

Disponibili: 10

Stato: Disponibile

Salatini Mix

€ 1,00

Disponibili: 10

Stato: Disponibile

Crostini

€ 0,90

Disponibili: 10

Stato: Disponibile

Mandorle Salate

€ 0,90

Disponibili: 10

Stato: Disponibile

Chipster

€ 1,00

Disponibili: 10

Stato: Disponibile

Snack al Formaggio

€ 1,50

Disponibili: 10

Stato: Disponibile

Barretta Milka

€ 0,90

Disponibili: 10

Stato: Disponibile

Barretta ai Cereali

€ 1,00

Disponibili: 10

Stato: Disponibile

Cioccolato Fondente

€ 1,50

Disponibili: 10

Stato: Disponibile

Wafer alla Nocciola

€ 1,10

Disponibili: 10

Stato: Disponibile

KitKat

€ 1,10

Disponibili: 10

Stato: Disponibile

Barretta Pro

€ 0,80

Disponibili: 10

Stato: Disponibile

Distributore(ID=2, Tipo: "Cibi e Bevande Freddo", Città: "Pavia", Via: "Corso Cavour", N. Civico: "15", ID Inventario=2)

Storico Spedizioni

Modifica Pr...

Inserisci il nuovo prezzo:

Conferma

Sostituzione Prodotto

Seleziona un prodotto sostitutivo

Tarallini

€ 0,60

Seleziona

Patatine Bio

€ 0,65

Seleziona

Tuc Cracker

€ 1,00

Seleziona

FUNZIONALITÀ

Selezione fornitore

Carrello con i prodotti per il rifornimento

Storico spedizioni

Seleziona un fornitore

Seleziona un fornitore per: Aranciata 33cl

Fornitore: Gourmet Italia Srl

Città: Milano

Prezzo unitario: €0.35

Seleziona

Fornitore: Dolce & Salato Srl


Città: Pavia

Prezzo unitario: €0.40

Quantità

Seleziona

Carrello




Prodotto: Aranciata 33cl

Fornitore: Gourmet Italia Srl

Quantità: 5

Totale: €1.63

Rimuovi



Prodotto: Aranciata 33cl

Fornitore: Dolce & Salato Srl

Quantità: 3

Totale: €1.14

Rimuovi

Torna ai Prodotti

Concludi Ordine

Storico Spedizioni

ID Spedizione: 2ZEHW



ID Prodotto: 26

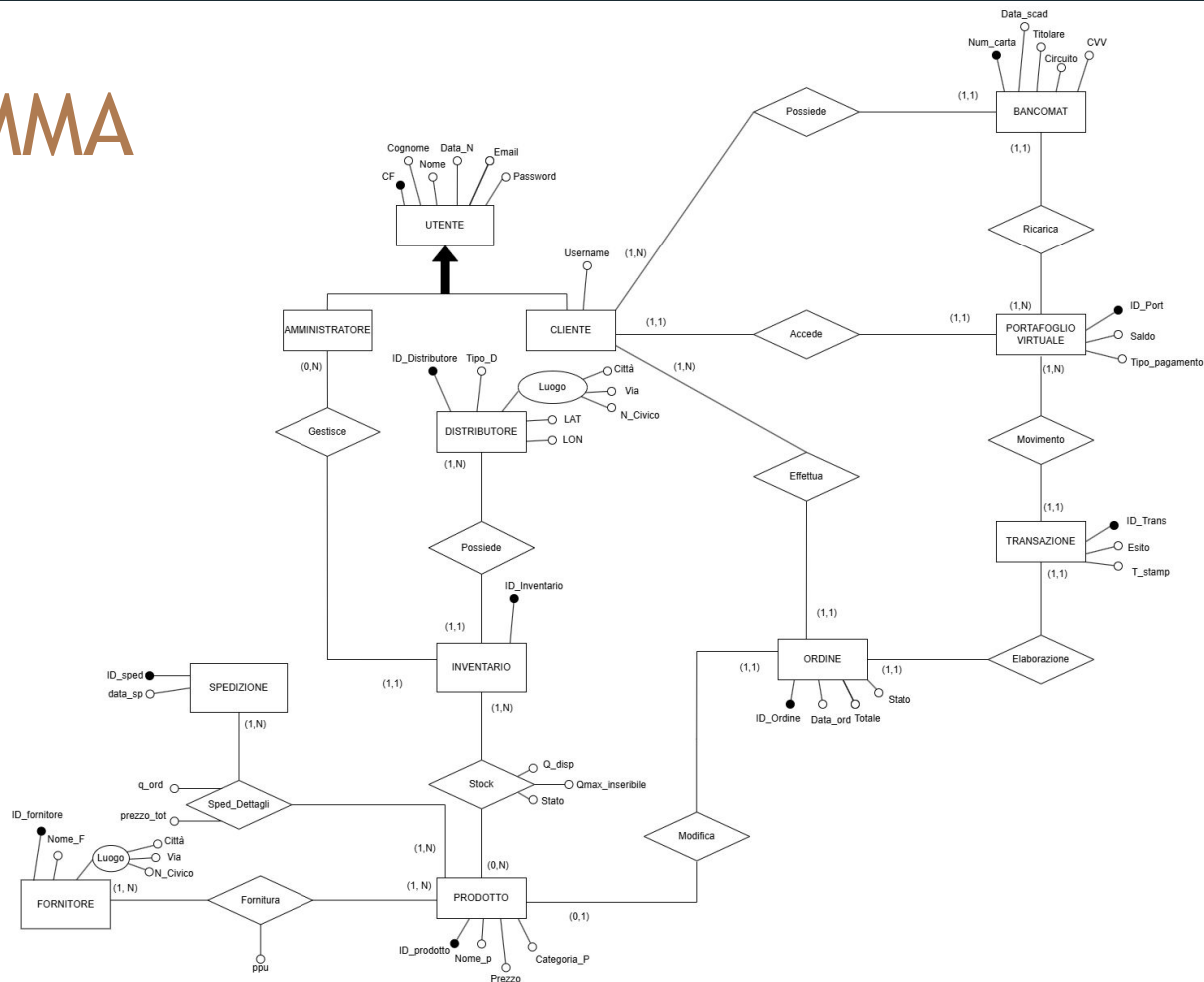
Quantità: 8

Prezzo Totale: €2.77

Data spedizione: 2025-02-27

Chiudi

DIAGRAMMA ERA



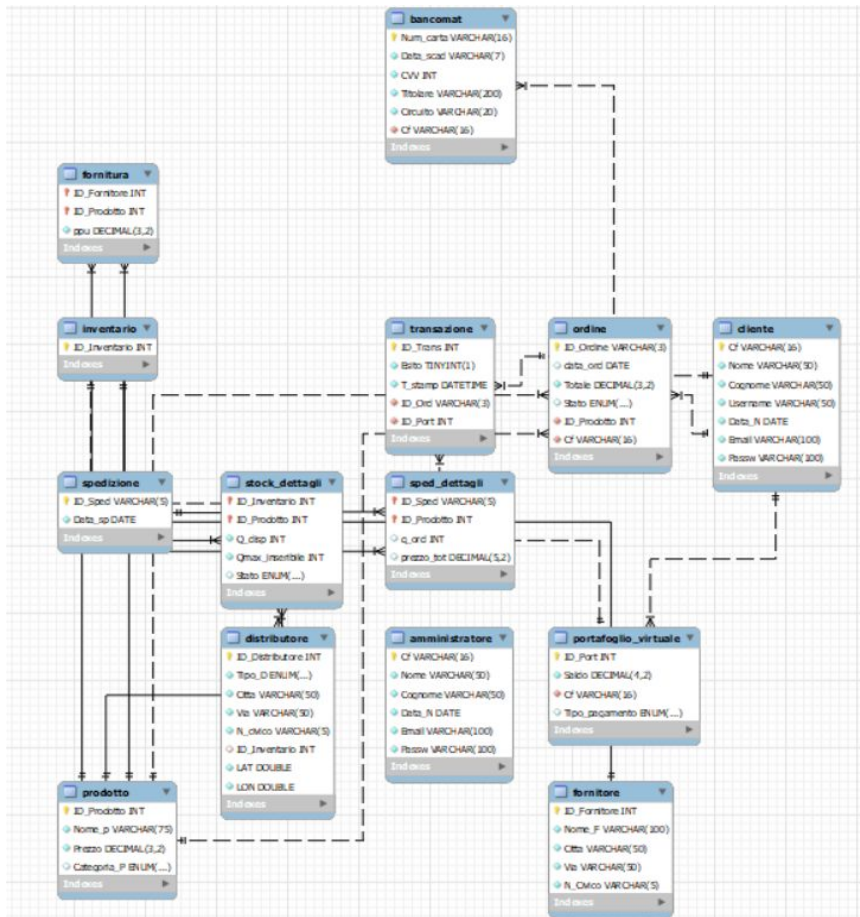
DATABASE

Il nostro sistema prevede continue interazioni con il database centralizzato per implementare correttamente gran parte delle funzionalità.

Il database è gestito tramite istanza locale, pertanto occorre importarlo adeguatamente sulla macchina desiderata per un corretto utilizzo del software .

Il software utilizzato per la sua realizzazione e configurazione è stato 'MySQLWorkbench CE 8.0'.

Di fianco, riportiamo il reverse engineer che descrive le table e le loro interazioni.



PROGETTAZIONE

Model-View-Controller MVC: Il nostro sistema è stato progettato seguendo il modello architetturale Model-View-Controller (MVC). Questo modello ci ha fornito una struttura chiara per separare i dati di dominio (Model), la logica di presentazione (View) e la logica di controllo (Controller), migliorando la modularità e la manutenibilità del nostro codice.

Data Access Object DAO: Abbiamo implementato elementi DAO per gestire l'accesso ai dati nel nostro sistema. Questi oggetti DAO forniscono un'interfaccia standard per interagire con la fonte dati sottostante e isolano la logica di accesso ai dati dal resto dell'applicazione, migliorando la scalabilità e la manutenibilità del nostro sistema

Abbiamo inoltre previsto un **Service Layer** che funge da intermediario tra i Controller e i Dao, centralizzando la logica di business e permettendo una maggiore separazione delle responsabilità.

GRAZIE PER L'ATTENZIONE

Gruppo: JARvisCrew