

# Esercizi di Ingegneria del Software

Il caso della “Grande Distribuzione”

*V. Ambriola, C. Montangelo e L. Semini*

Corso di Laurea in Informatica  
Corso di Laurea in Informatica Applicata

Dipartimento di Informatica  
Università di Pisa



© 2009

# INDICE

<b>IL PROBLEMA.....</b>	<b>4</b>
<b>1. ANALISI DEL DOMINIO.....</b>	<b>6</b>
1. Ripristino scorte con trasporto tra i punti vendita della catena (classi).....	6
2. Ripristino scorte con trasporto tra i punti vendita della catena (attività).....	6
3. Ripristino scorte con trasporto tra i punti vendita della catena (macc. a stati).....	7
4. Gestione della cassa per gli acquisti (classi). ....	8
5. Modifica dei prezzi (classi).....	9
<b>2. ANALISI DEI REQUISITI.....</b>	<b>10</b>
6. Modifica dei prezzi di vendita (caso d'uso). ....	10
7. Casi d'uso multipli.....	10
<b>3. ARCHITETTURA.....</b>	<b>13</b>
8. Aggiornamento dei prezzi (C&C).....	13
9. Aggiornamento dei prezzi (Dislocazione). ....	13
10. Gestione cassa acquisti (Strutturale). ....	14
11. Trading system. ....	15
<b>4. PROGETTAZIONE DI DETTAGLIO.....</b>	<b>17</b>
12. GestionePuntoVendita (struttura). ....	17
13. GestionePagamenti (struttura).....	17
14. Aggiornamento dei prezzi (dinamica).....	18
15. Modifica dei prezzi (realizzazione caso d'uso).....	19
16. InvioOrdini (Realizzazione Casi d'uso).....	20
17. Cassa per gli acquisti (Realizzazione casi d'uso). ....	21
18. Modifica dei prezzi (Architettura di dettaglio).....	22
<b>5. VERIFICA DEL SOFTWARE.....</b>	<b>25</b>
19. Modifica dei prezzi. ....	25

20.	Cassa per gli acquisti.....	27
21.	Modifica dei prezzi (Ambiente).....	28

## **Descrizione del problema.**

L'esempio descrive un sistema per il commercio, chiamato TradingSystem, come lo si può osservare in un supermercato. Comprende sia i processi alla cassa, tipo la scansione dei prodotti usando un lettore di codice a barre e il pagamento con carta di credito o contanti, sia compiti di gestione quali l'ordine di prodotti quando si stanno esaurendo le scorte o la generazione di resoconti.

La cassa è il posto dove il cassiere scandisce i prodotti che il cliente vuole comprare e dove avviene il pagamento, in contanti, con bancomat, o con carta di credito. Una cassa può cambiare modalità da normale a rapida, e accettare solo clienti con un limitato numero di prodotti, e pagamento in contanti. Per gestire i processi alla cassa sono necessari alcuni dispositivi hardware: un PC, un registratore di cassa, un display, una stampante (per le ricevute di bancomat e della carta di credito), un lettore di carte con tastierino e un lettore di codici a barre.

Il registratore di cassa permette di iniziare e finire un acquisto, e di gestire un pagamento in contanti. Per i pagamenti con carta di credito o bancomat si usa il lettore di schede. Il cassiere identifica tutti gli articoli che il cliente intende comprare con il lettore di codici a barre. Alla fine del pagamento, si produce lo scontrino con la stampante. Ogni cassa ha un display che segnala se la cassa sta operando in modo "espresso" o normale. Il cassiere può tornare al modo normale da quello espresso premendo un apposito pulsante.

L'unità centrale di ogni cassa è il PC di cassa, che connette tutti gli altri componenti. Anche il software responsabile della gestione del processo di vendita, ivi comprese le comunicazioni con la banca, gira su questa macchina.

Un punto vendita contiene diverse casse, collettivamente chiamate barriera casse. La barriera casse è collegata al ServerDelPuntoVendita, a sua volta collegato al ClientDelPuntoVendita. Il ServerDelPuntoVendita mantiene l'inventario del corrispondente punto vendita. Il ClientDelPuntoVendita viene usato dal responsabile del punto vendita per modificare i prezzi di vendita (a negozio chiuso) e per fare gli ordini. La modifica di un prezzo si riflette automaticamente nella modifica del prezzo esposto su un piccolo display posto sugli scaffali dove sono esposti i prodotti. Quando dei prodotti arrivano al punto vendita, il responsabile controlla la corrispondenza rispetto all'ordine fatto prima di inserirli nell'inventario. Inoltre, il TradingSystem permette al responsabile del punto vendita di generare resoconti sullo stato del magazzino.

Un insieme di punti vendita è organizzato in una catena di grande distribuzione: un ServerDellaCatena è collegato a tutti i punti vendita. Usando un ClientDellaCatena, il responsabile della catena può generare diversi tipi di resoconti. Ad esempio, TradingSystem permette di calcolare i tempi medi di consegna da un fornitore ai punti vendita.

Se un negozio finisce le scorte di un prodotto, viene mandata una query per sapere se il prodotto è disponibile in qualche altro punto vendita della catena. A questo punto, il ServerDellaCatena deve sincronizzarsi con i ServerDiPuntoVendita per avere dati globali aggiornati. Dopo che una query ha avuto successo, i prodotti richiesti sono inviati da un punto vendita ad un altro. Deve però essere valutato, con l'uso di euristiche, se il trasporto è utile. Per esempio, se il prodotto scarseggia in tutti i punti vendita e si presume finirà prima della successiva fornitura, non ha senso effettuare il trasporto. Una volta che il trasporto è stato deciso, la quantità scambiata è inserita come "in arrivo" al punto vendita ricevente. Quando la merce arriva al punto vendita, viene rimossa l'etichetta "in arrivo".

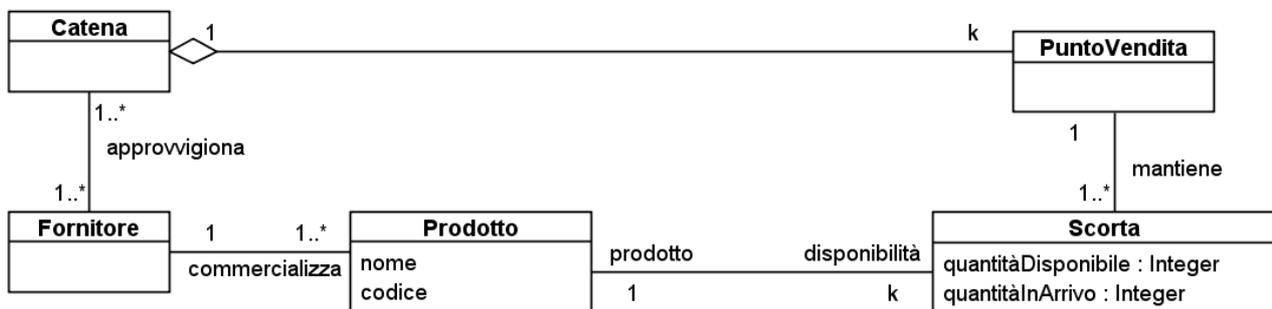
# 1. Analisi del dominio

## 1. Ripristino scorte con trasporto tra i punti vendita della catena (classi).

Si assuma che la catena sia formata da  $k$  punti vendita.

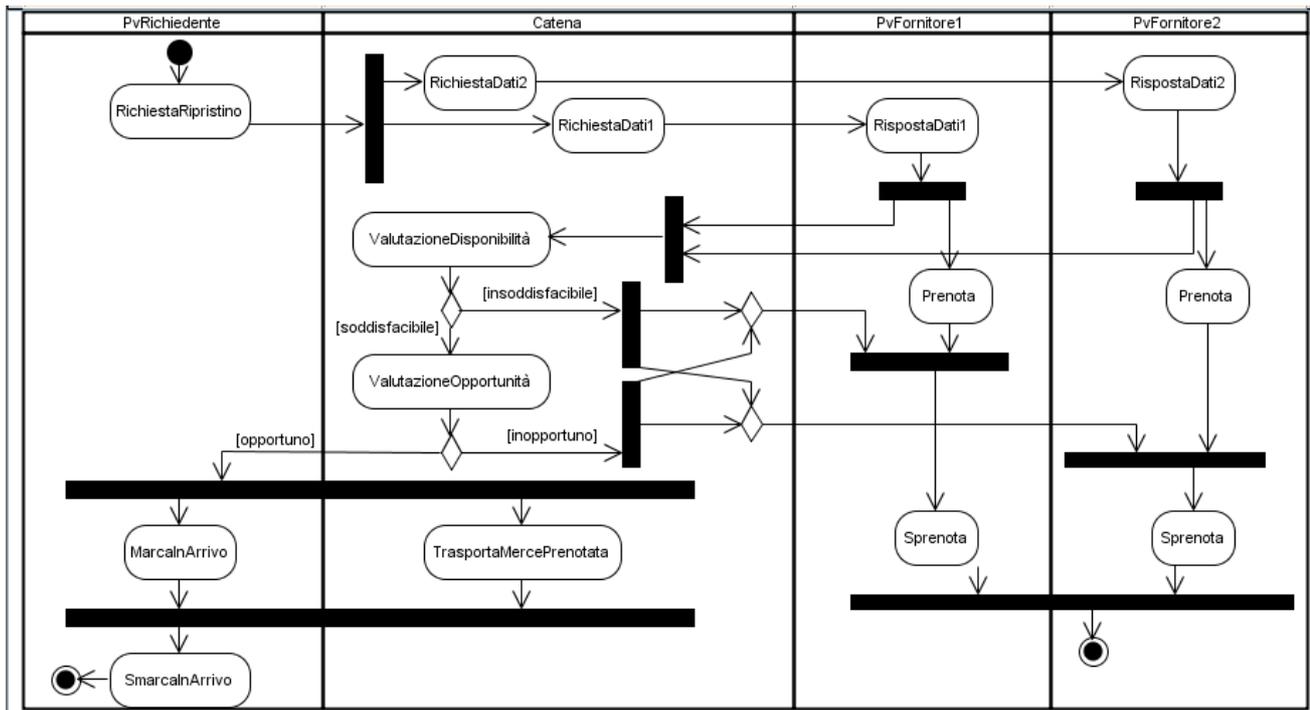
Dare un diagramma delle classi, considerando come classi o attributi tutti e soli i termini seguenti: prodotto, nome, codice, scorta, quantitàDisponibile, quantitàInArrivo, PuntoVendita, Catena, Fornitore.

Un possibile diagramma delle classi è il seguente:



## 2. Ripristino scorte con trasporto tra i punti vendita della catena (attività).

Dare un diagramma di attività che descrive il processo di ripristino scorte con trasporto tra i punti vendita della catena, che avviene quando uno dei punti vendita ha esaurito le scorte di un prodotto. Si assuma che la catena sia formata da 3 punti vendita e si usino le partizioni per attribuire la responsabilità delle azioni ai vari punti vendita e alla catena.



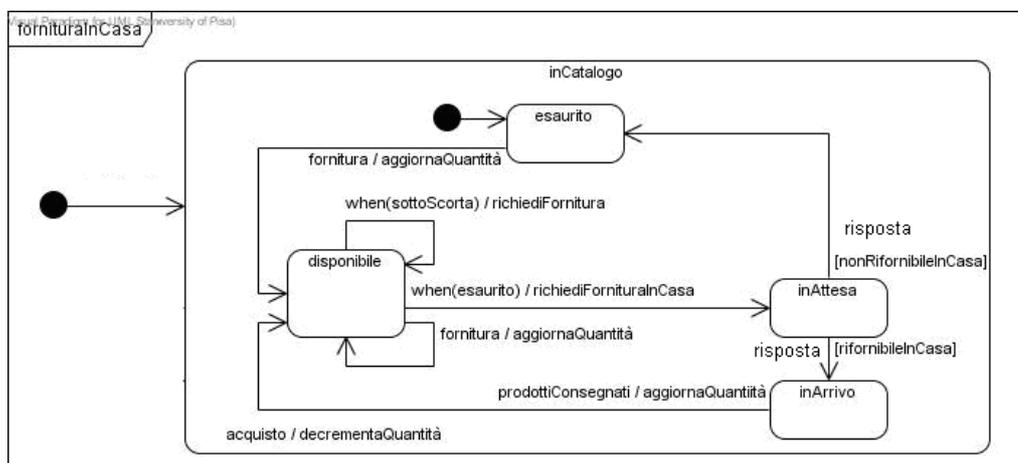
Nel diagramma precedente si è tenuto conto che:

l'attività di valutazione dell'opportunità del trasporto tra punti vendita possa essere sufficientemente lunga da introdurre la prenotazione della quantità di articoli da trasportare, per evitare che vengano venduti nel frattempo.

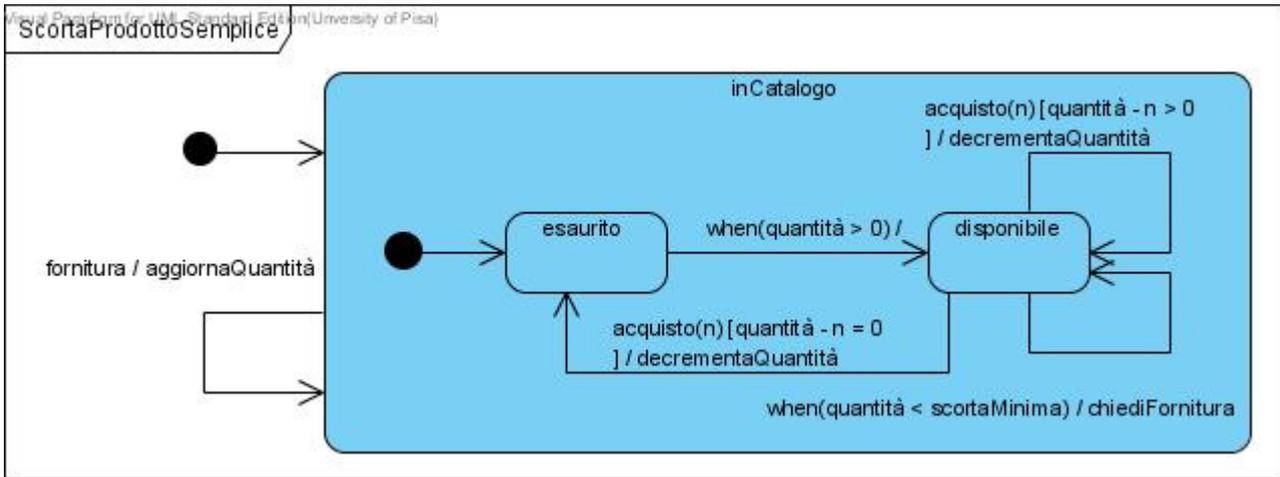
### 3. Ripristino scorte con trasporto tra i punti vendita della catena (macc. a stati).

Dare un diagramma di macchina a stati che descriva l'evoluzione della scorta di un prodotto, tenendo conto degli acquisti, del normale ripristino delle scorte quando la quantità disponibile scende sotto una data soglia, e della possibilità di trasferire articoli da un punto vendita all'altro, se un prodotto è esaurito.

Un possibile diagramma di macchina a stati è il seguente:



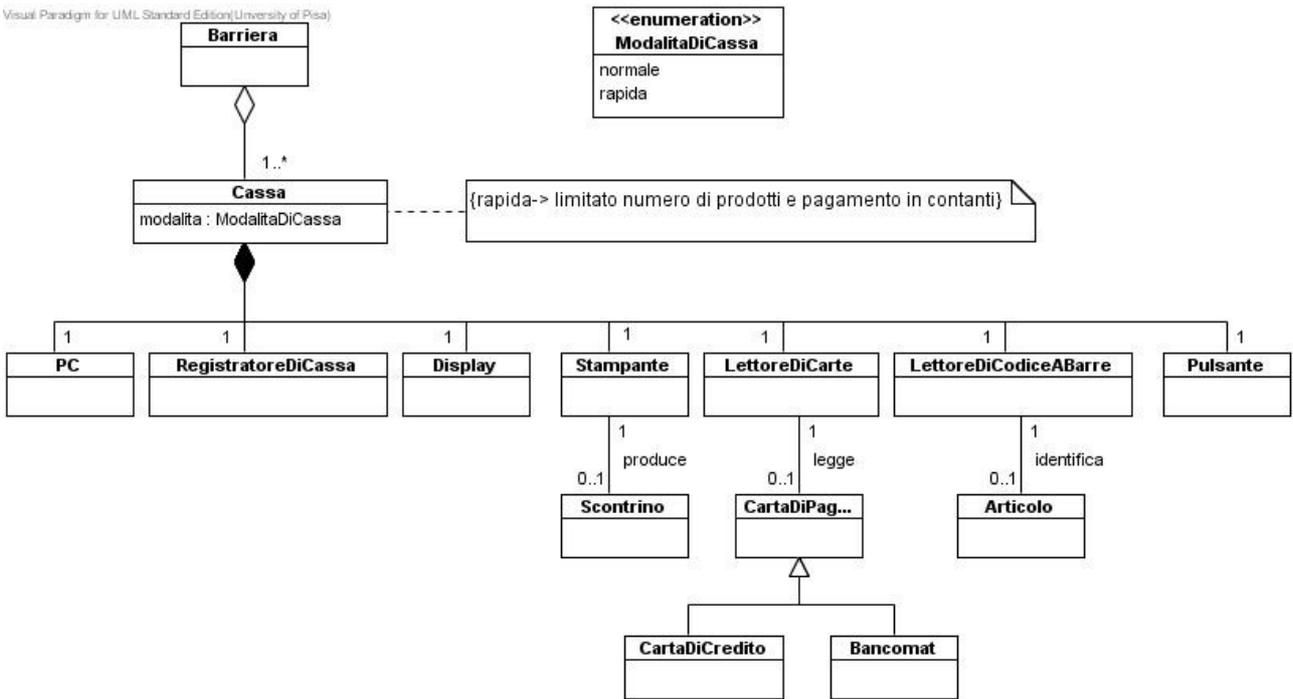
Se si ignora la possibilità di trasferire articoli da un punto vendita all'altro si ha un diagramma diverso.



#### 4. Gestione della cassa per gli acquisti (classi).

Dare un diagramma delle classi, considerando come classi, attributi o costanti tutti e soli i termini seguenti: articolo, bancomat, barriera, carta di credito, carta di pagamento, cassa, display, lettore di carte, lettore di codice a barre, modalità, normale (modalità di cassa), PC, pulsante di commutazione modalità di cassa, rapida (modalità di cassa), registratore di cassa, scontrino, stampante.

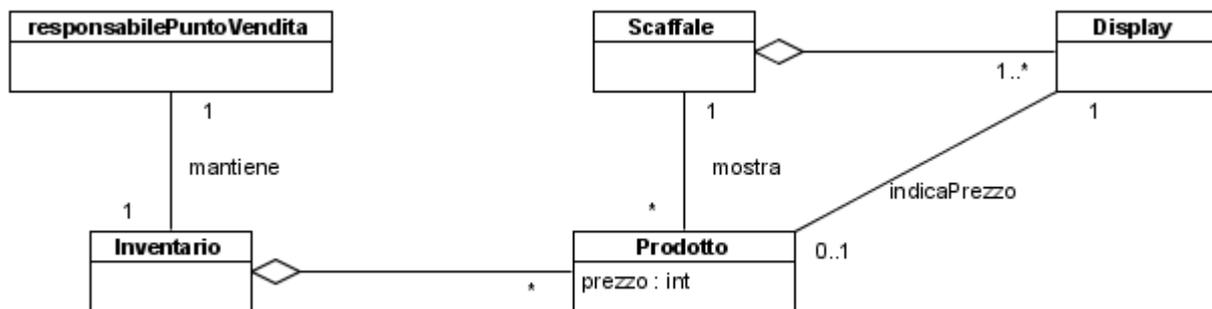
**Risposta.**



## 5. Modifica dei prezzi (classi).

Dare un diagramma delle classi considerando come classi e/o attributi: scaffale, display, responsabilePuntoVendita, inventario, prodotto, prezzo.

### Soluzione.



## 2. Analisi dei requisiti

### 6. Modifica dei prezzi di vendita (caso d'uso).

Dare la narrativa del caso d'uso **ModificaDeiPrezzi**, tenendo conto che la politica della catena è che le modifiche dei prezzi possono essere fatte una sola volta nelle 24 ore e non possono superare il 20 per cento.

#### **Risposta.**

Caso d'uso: **ModificaDeiPrezzi**

Breve descrizione: *Permette di modificare i prezzi di vendita di alcuni prodotti e la loro visualizzazione sul display allo scaffale.*

Attore principale: *ResponsabilePuntoVendita*

Attore secondario: *Nessuno*

PreCondizioni: *Punto vendita chiuso*

PostCondizioni: *Prezzi aggiornati*

Sequenza principale degli eventi:

- 1 *per (ogni prezzo da aggiornare)*
  - 1.1 *il Responsabile richiama il prodotto*
  - 1.2 *il Responsabile indica il nuovo prezzo*
  - 1.3 *il Sistema aggiorna la scheda del prodotto*
  - 1.4 *il Sistema aggiorna il display*

Sequenza alternativa degli eventi: **Variazione di prezzo superiore al 20% per almeno un articolo.**

Breve descrizione: *Gli articoli in questione vengono modificati in più o meno 20%.*

Sequenza alternativa degli eventi: **Prezzo di almeno un articolo già modificato nelle 24 ore.**

Breve descrizione: *Il prezzo degli articoli in questione non viene modificato e il Responsabile viene avvertito per ognuno.*

### 7. Casi d'uso multipli

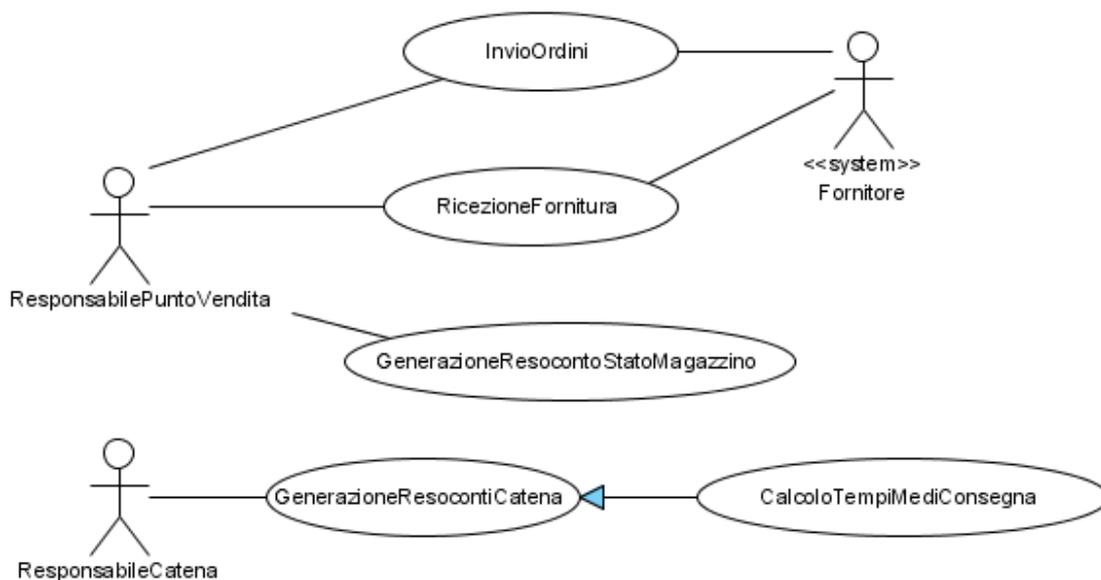
InvioOrdini, RicezioneFornitura, GenerazioneResocontoStato-Magazzino, e CalcoloTempiMedi-Consegna.

a. Dare un diagramma dei casi d'uso che comprenda i casi d'uso sopra elencati.

- b. Dare la narrativa del caso d'uso **InvioOrdini**, assumendo che un ordine venga fatto solo se non è possibile il trasferimento interno

**Risposta:**

a.



b.

### Caso d'uso: **InvioOrdini**

Breve descrizione: *Permette di inviare un ordine di alcuni prodotti a dei fornitori*

Attore principale: *ResponsabilePuntoVendita*

Attore secondario: *Fornitore*

PreCondizioni: *Prodotti sotto scorta, impossibile il trasferimento interno*

PostCondizioni: *Prodotti ordinati*

Sequenza principale degli eventi:

1. *per (ogni fornitore)*

1.1. *il ResponsabilPuntoVendita chiede al TradingSystem di creare un nuovo ordine per il fornitore*

1.2. *per (ogni prodotto da ordinare a quel fornitore)*

1.2.1. *il ResponsabilPuntoVendita chiede al TradingSystem di aggiungere il prodotto all'ordine*

1.3. *il Responsabile chiede a TradingSystem l'invio dell'ordine*

Sequenza alternativa degli eventi:

*Fornitore non più convenzionato con la catena  
Prodotto non più in commercio.*

### 3. Architettura

#### 8. Aggiornamento dei prezzi (C&C).

Fornire la vista C&C considerando la seguente descrizione per le componenti, e mostrare l'interfaccia che GestionePuntoVendita deve offrire per realizzare il caso d'uso.

Componente	Responsabilità
DBPuntoVendita	Memorizza i dati sui prodotti in vendita.
GestionePuntoVendita	Gestisce l'aggiornamento dei prezzi del punto vendita.
ControlloPuntoVendita	Permette al ResponsabilePuntoVendita di accedere al sistema per aggiornare i prezzi del punto vendita.
DisplayPrezzo	Gestisce la visualizzazione dei prezzi sugli scaffali.

#### Risposta.

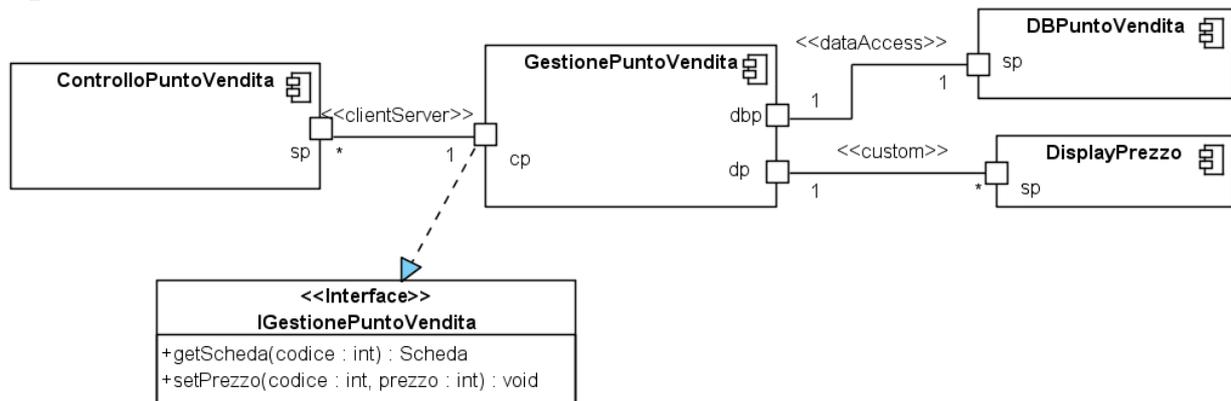
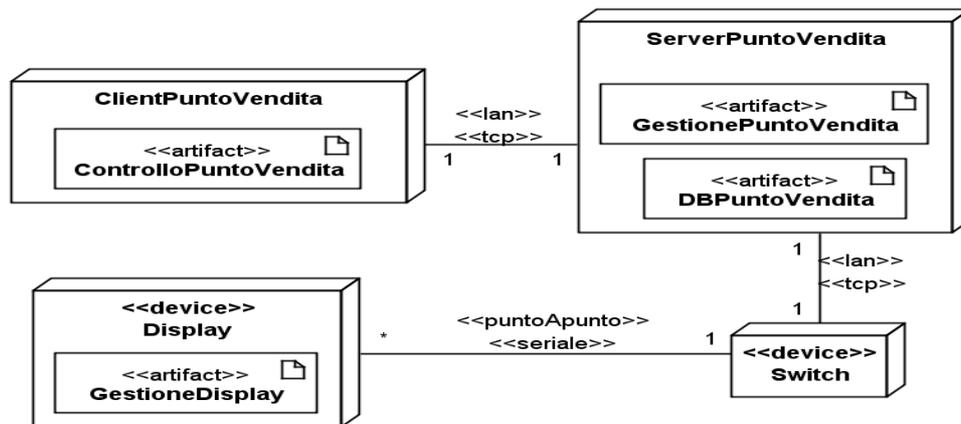


Figura 1 Architettura modifica prezzi

#### 9. Aggiornamento dei prezzi (Dislocazione).

Fornire la vista di dislocazione. Si tenga conto che, oltre ai nodi hardware descritti nel testo, è previsto uno switch per inoltrare i messaggi ai Display, connesso via rete locale al Server e via cavo ai singoli Display. Il software dello switch è reso disponibile dal produttore, e non interessa in questa vista.

#### Risposta.



## 10. Gestione cassa acquisti (Strutturale).

La vista C&C dell'architettura proposta per realizzare la gestione di cassa è la seguente:

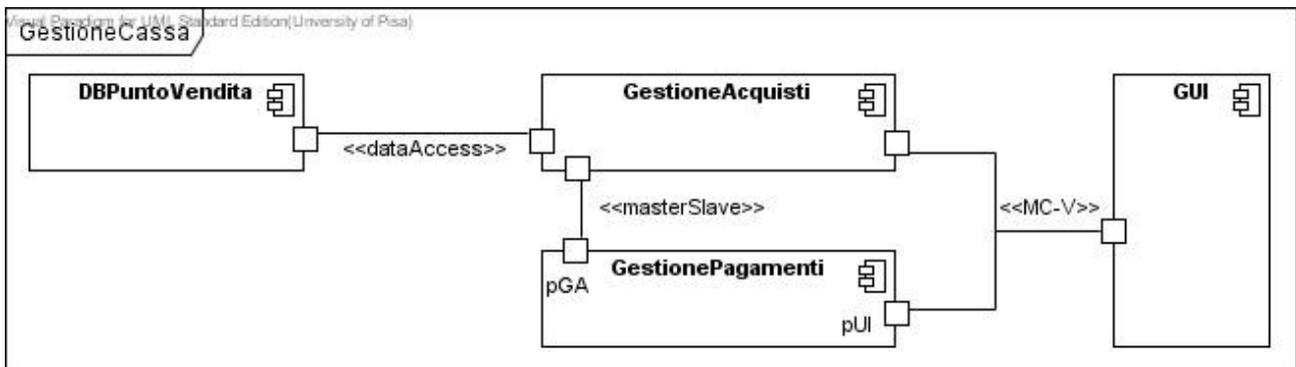


Figura 2 Architettura gestione cassa

In questo modello, il connettore <<masterSlave>> denota una interazione di tipo cliente-servente, ma con un solo cliente (GestioneAcquisti, in questo caso), e <<MC-V>> l'interazione tra la componente di visualizzazione in un'interfaccia grafica e il controllore dell'interfaccia stessa.

Le componenti hanno le seguenti responsabilità:

Componente	Responsabilità
Gestione Acquisti	Gestisce l'interazione col cassiere per il calcolo del totale della spesa, la stampa dello scontrino, la scelta della forma di pagamento e invoca GestionePagamenti.
GestionePagamenti	Gestisce l'interazione con il cassiere, il cliente e la banca,

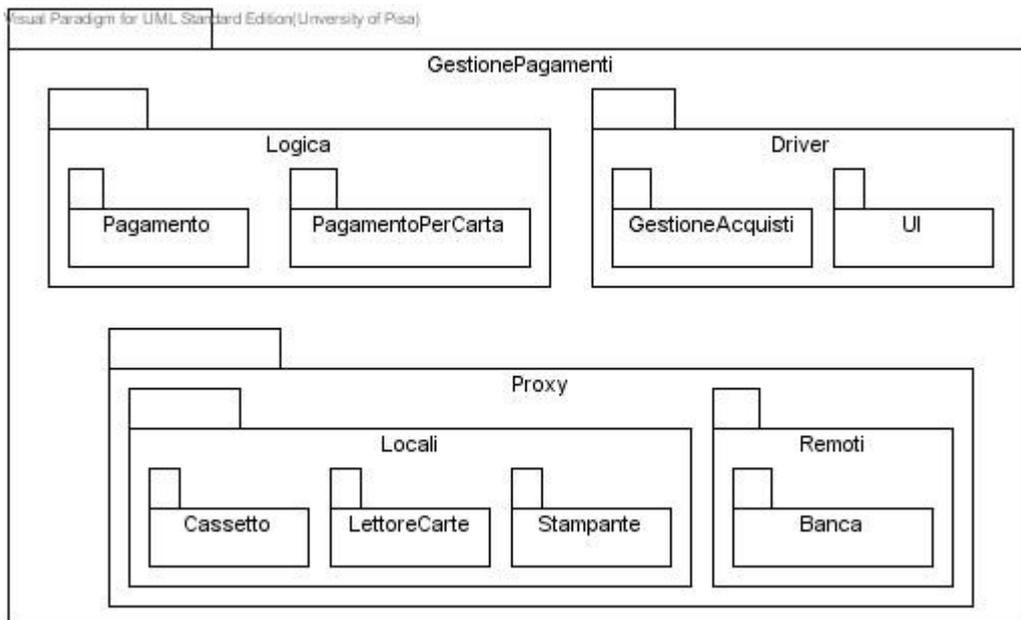
	secondo le varie forme di pagamento, attraverso le apparecchiature di volta in volta necessarie (cassetto, lettore di carte, ecc.).
GUI	Gestisce la parte grafica delle interazioni.
DBPuntoVendita	Contiene le informazioni sugli articoli in vendita.

Si noti che, nella vista data sopra, si sono implicitamente considerate come esterne al sistema da realizzare tutte le apparecchiature: la stampante, il lettore di carte, ecc.

Fornire una vista strutturale di decomposizione del modulo GestionePagamenti, in base alle seguenti regole:

- il massimo livello di decomposizione coincide con le parti indicate nella soluzione all'esercizio precedente;
- tutte le parti devono essere considerate;
- la struttura della decomposizione deve separare logica, driver e proxy.
- la struttura della decomposizione deve separare i proxy verso sistemi esterni remoti da quelli verso le apparecchiature locali.

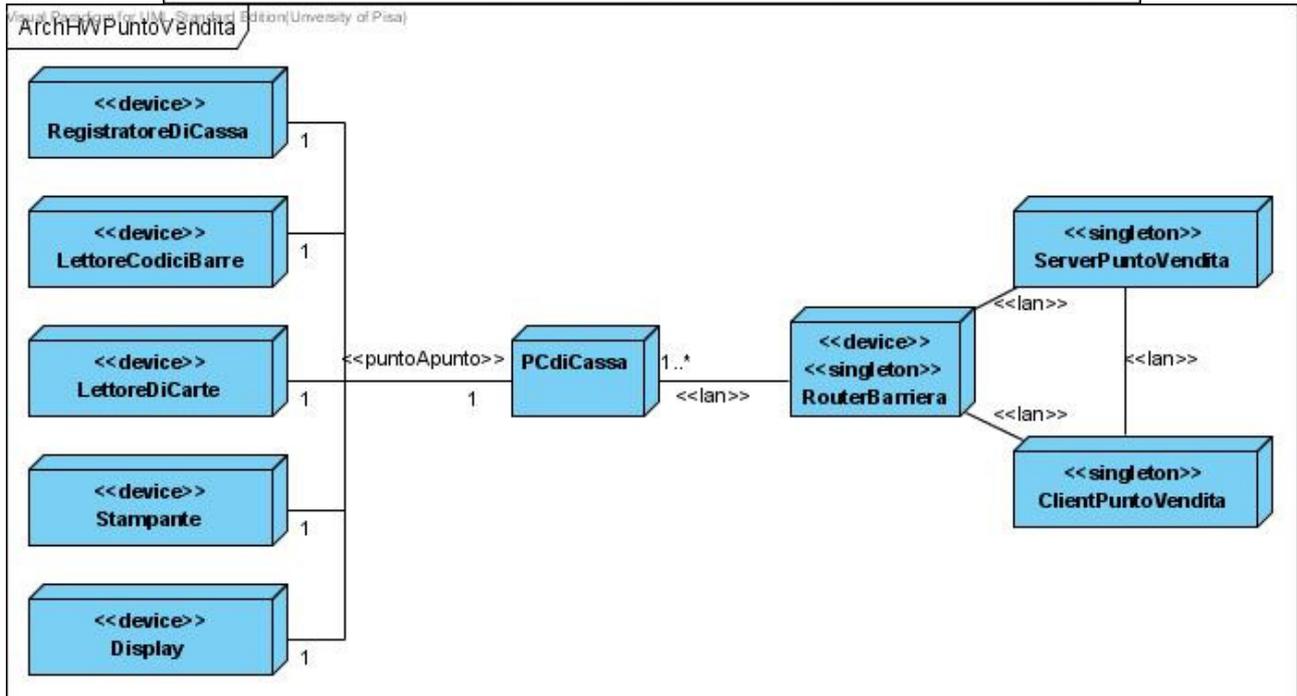
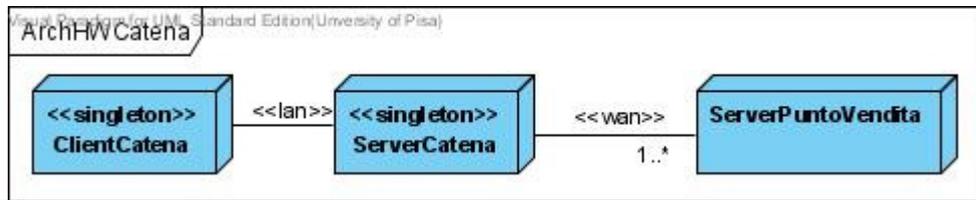
### Risposta.



## 11. Trading system.

Dare un diagramma di deployment parziale, che consideri solo la struttura hardware di TradingSystem. Si assuma l'esistenza di un router che funge da gateway per il sotto-sistema barriera.

**Risposta:** Si vedano i prossimi due diagrammi, che danno il punto di vista della catena e del punto vendita, rispettivamente.

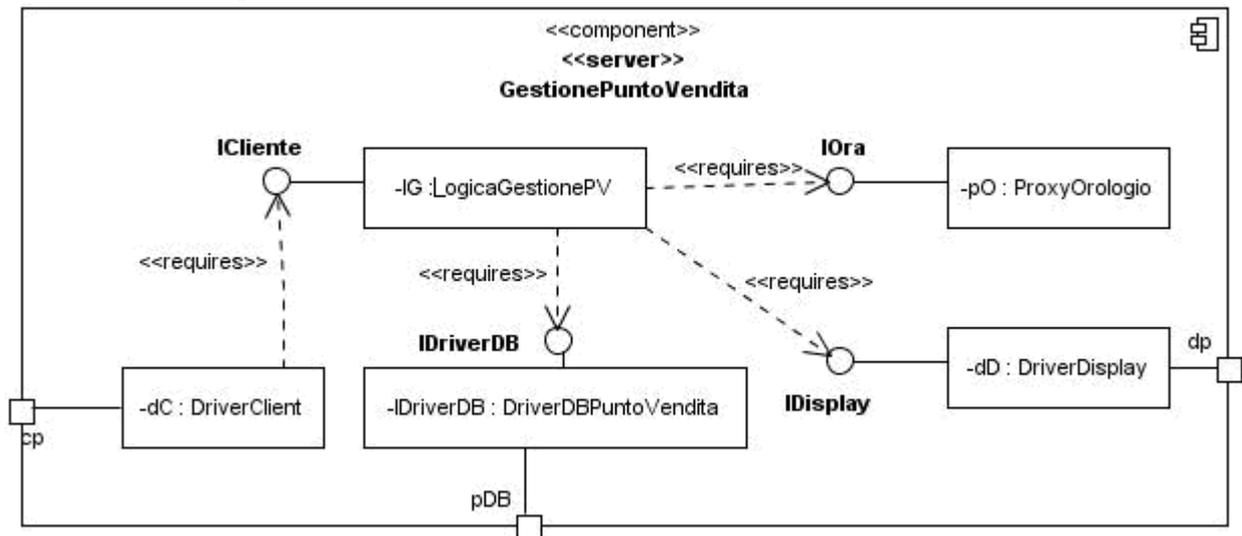


## 4. Progettazione di dettaglio

### 12. GestionePuntoVendita (struttura).

Fornire il diagramma di struttura composita della componente GestionePuntoVendita in Figura 1. Si considerino esplicitamente anche le interazioni con l'orologio, necessarie per evitare doppie modifiche di prezzo nelle 24 ore.

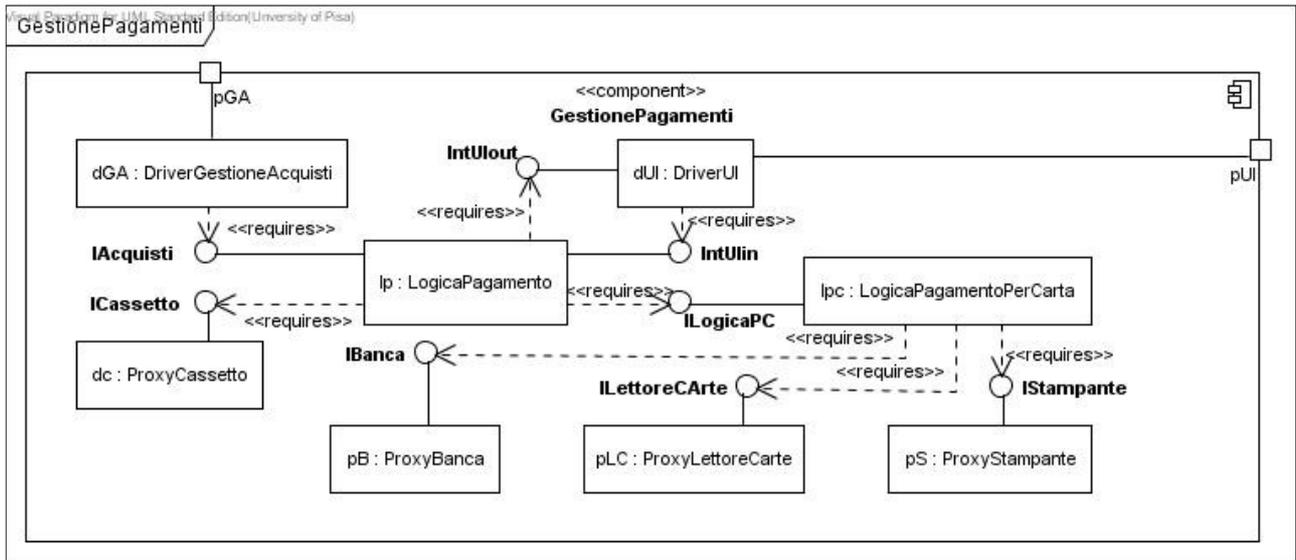
**Risposta.**



### 13. GestionePagamenti (struttura).

Dare un diagramma di struttura composita della componente GestionePagamenti in Figura 2. Si distingua tra la logica di pagamento e quella di pagamento con carte.

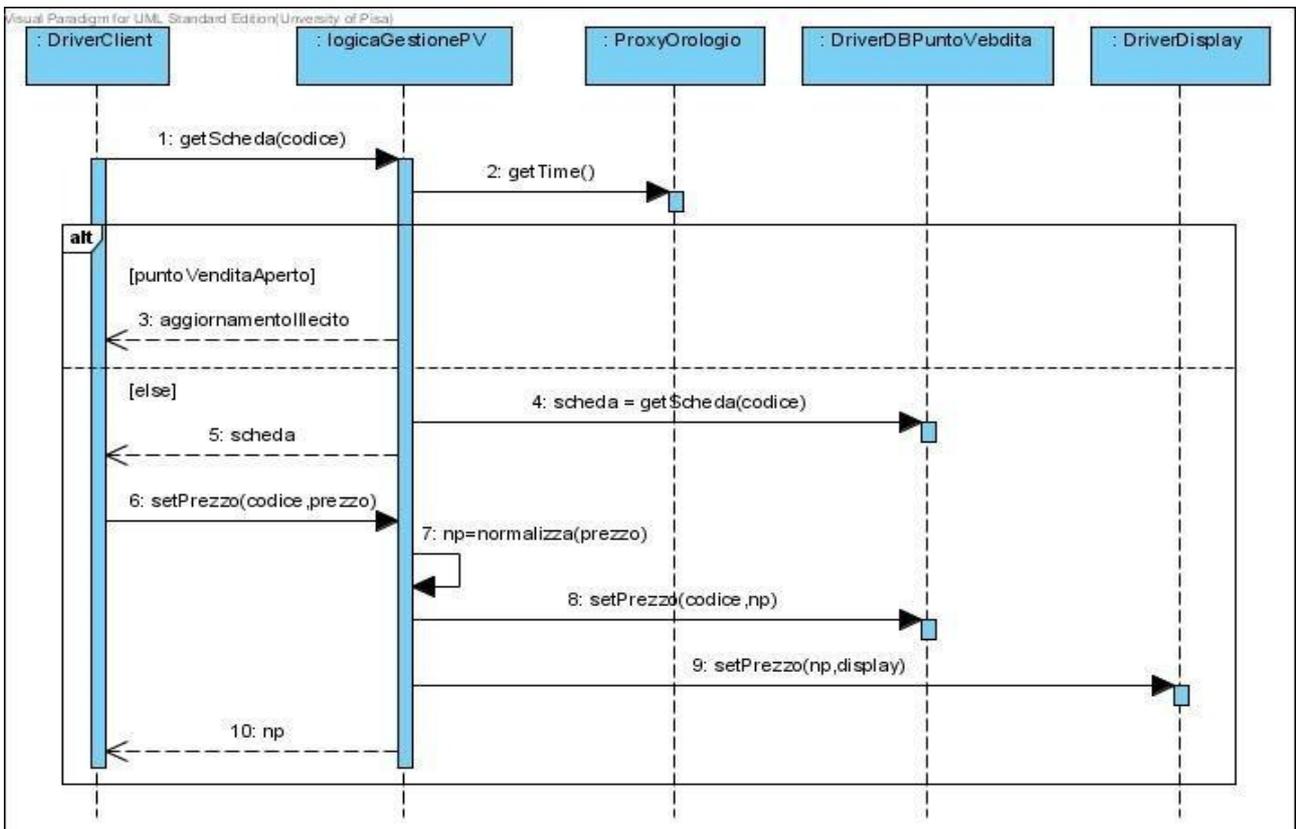
**Risposta.**



#### 14. Aggiornamento dei prezzi (scambiare con ex. 15).

Fornire un diagramma di sequenza che mostri come le parti della componente GestionePuntoVendita cooperino per realizzare il caso d'uso ModificaDeiPrezzi. Includere un controllo sul soddisfacimento della precondizione, e la descrizione della prima sequenza alternativa degli eventi. .

**Risposta.**

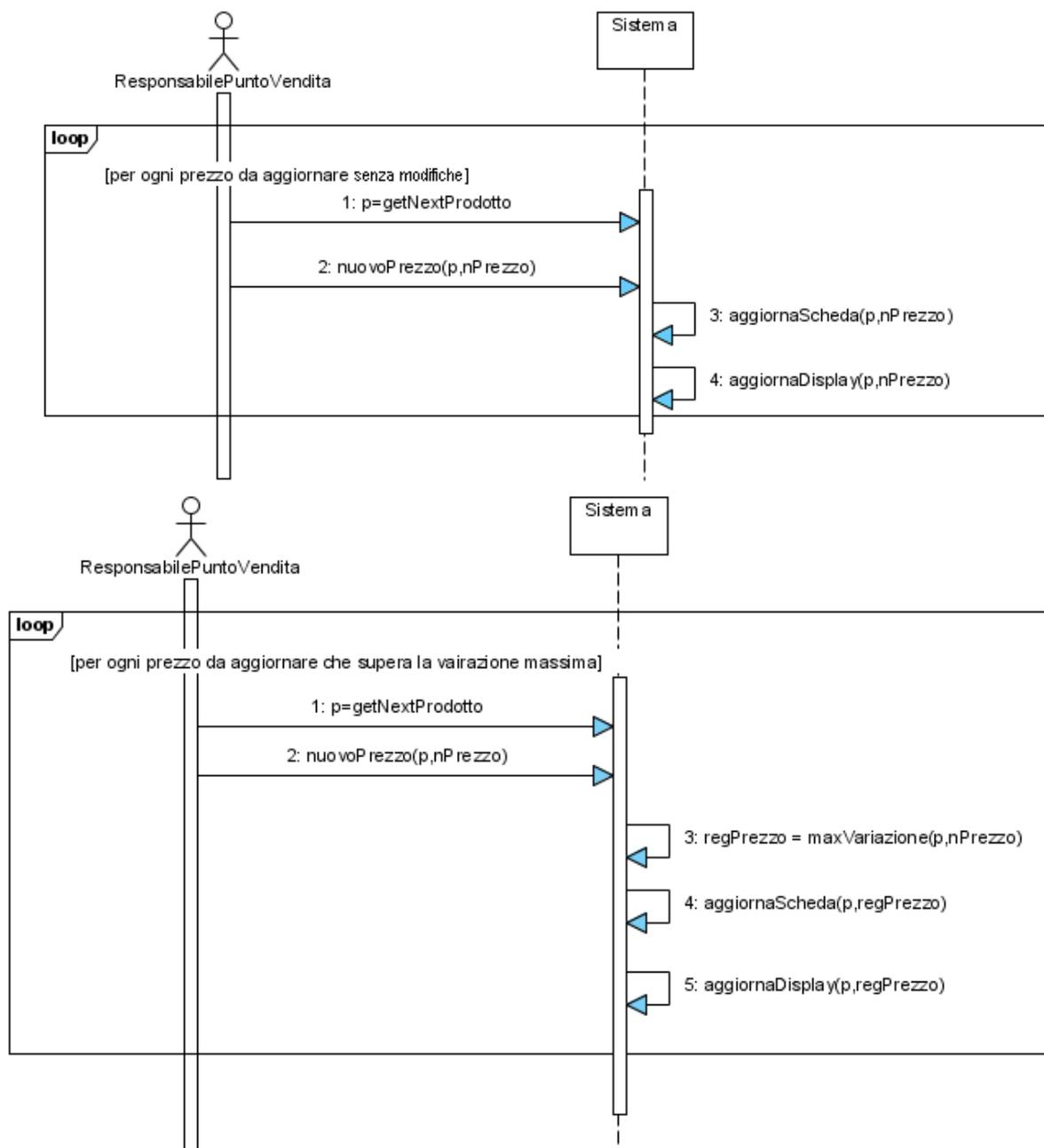


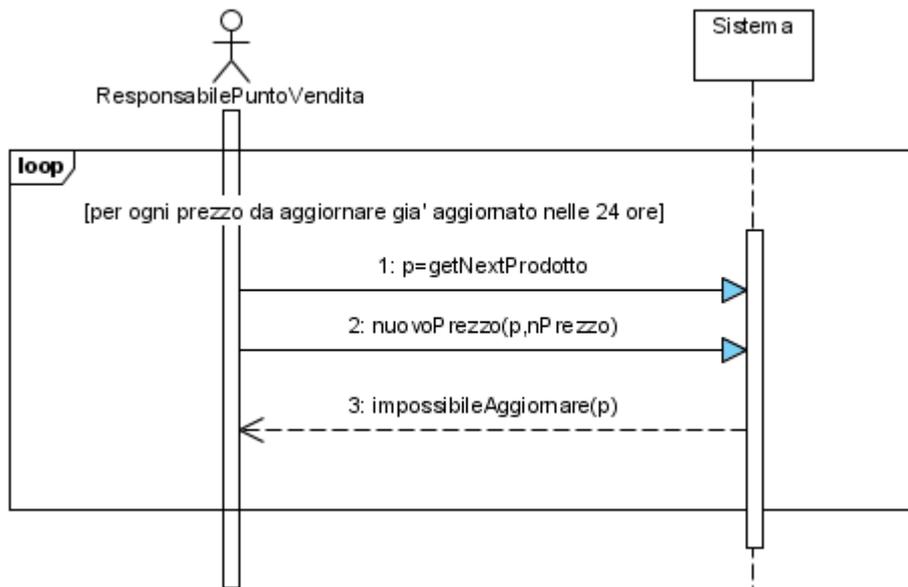
## 15. Modifica dei prezzi (realizzazione caso d'uso).

Descrivere, utilizzando i diagrammi di sequenza, il caso d'uso **ModificaDeiPrezzi**.

### Soluzione

Come esempio, mostriamo 3 diagrammi separati, lasciando all'analista il compito di integrarli in un diagramma unico.

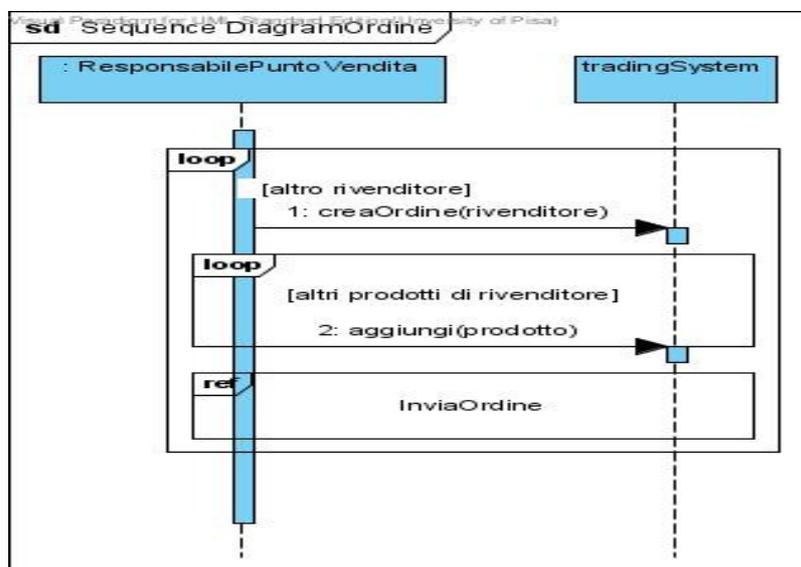




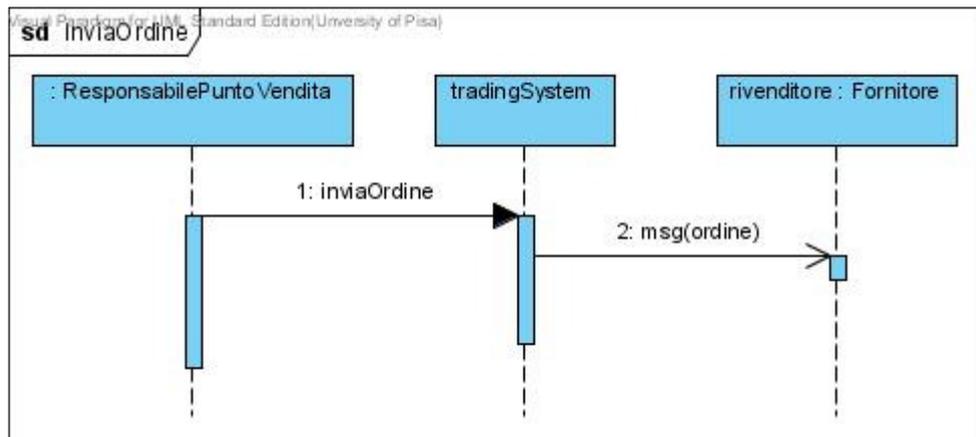
## 16. InvioOrdini (Realizzazione Casi d'uso)

Dare, con uno o più diagrammi di sequenza, la sequenza principale degli eventi del caso d'uso InvioOrdini.

**Risposta:** Una possibile soluzione è



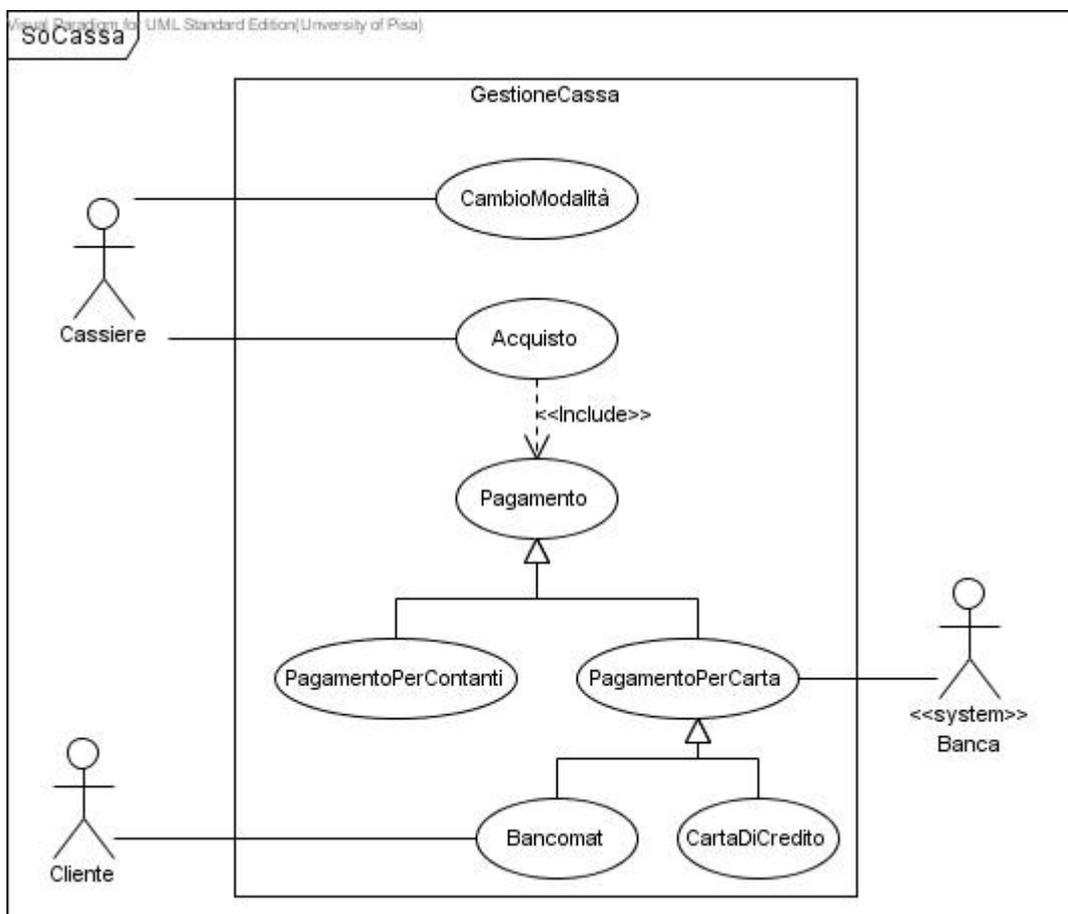
dove il frammento combinato InviaOrdine considera il rivenditore corrente nel ciclo esterno:



N.B. Il frammento riferito ha un parametro, ossia il particolare fornitore. Mettere la linea di vita nel primo diagramma non avrebbe espresso correttamente questa dipendenza dall'indice del loop.

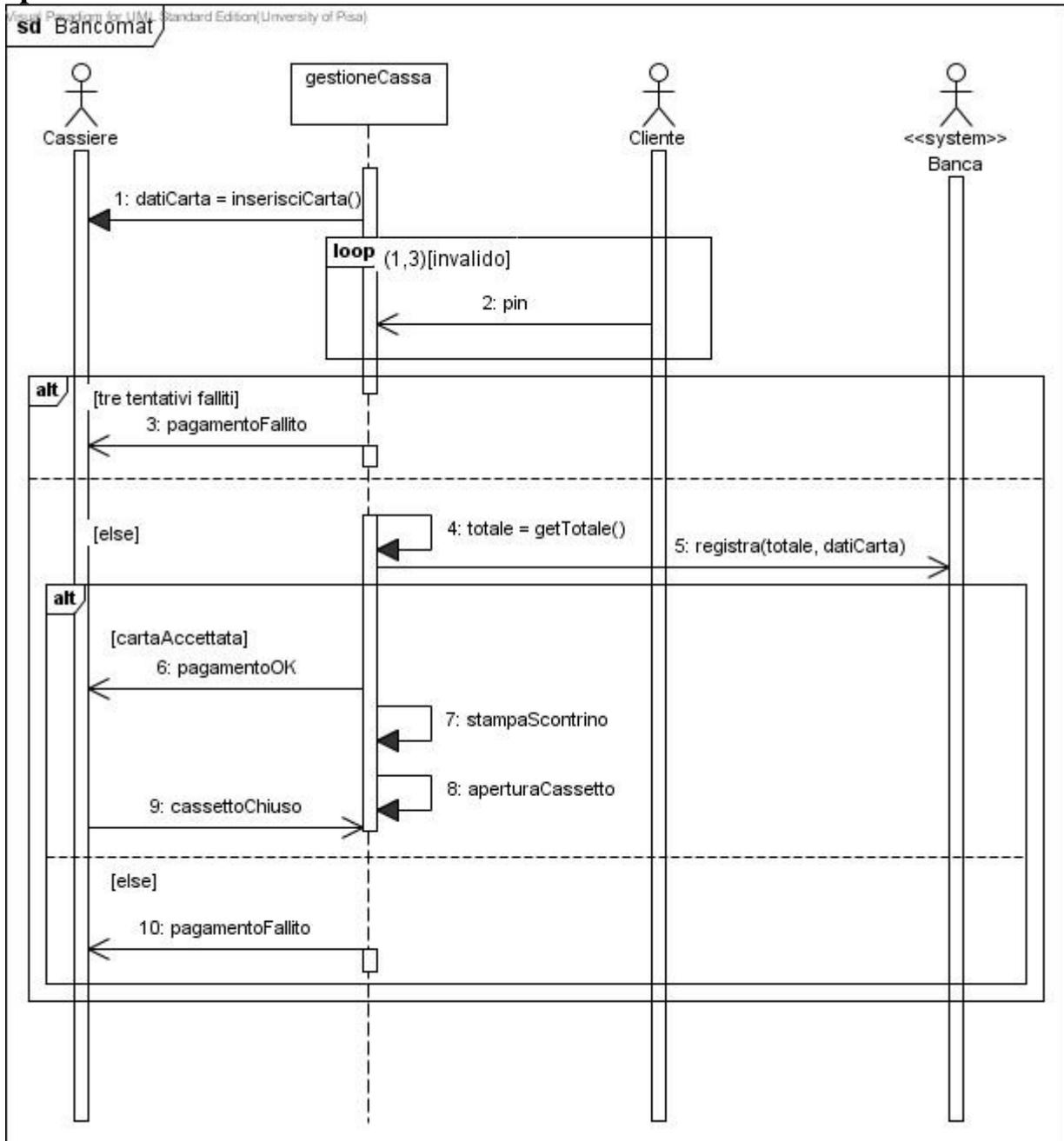
### 17. Cassa per gli acquisti (Realizzazione casi d'uso).

Per il sottosistema GestioneCassa sono stati individuati come prioritari i casi d'uso per cambiare modalità di cassa e per l'acquisto, che comprende le varie forme di pagamento, come specificato nel seguente diagramma:



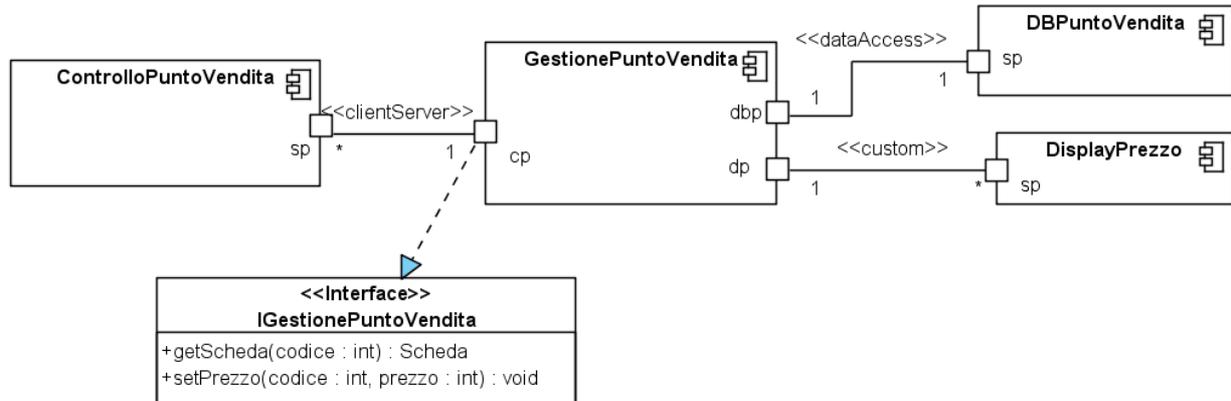
Dare, con un diagramma di sequenza, la specifica del caso d'uso pagamento con bancomat. Si assumano le seguenti operazioni interne di GestioneCassa: `getTotale` (per ottenere il totale da pagare), `stampaScontrino` (per la stampa delle ricevute) e `apriCassetto` (per permettere al Cassiere di mettere nel cassetto della Cassa la ricevuta del Bancomat). La chiusura del cassetto viene vista come un messaggio del Cassiere al sistema.

**Risposta.**



18. Modifica dei prezzi (Architettura di dettaglio).

La fase di progettazione ha portato alla definizione della vista C&C della parte di architettura del sistema TradingSystem che permette l'aggiornamento dei prezzi, e alla definizione del diagramma di struttura composita della componente GestionePuntoVendita



Componente	Responsabilità
DBPuntoVendita	Memorizza i dati sui prodotti in vendita.
GestionePuntoVendita	Gestisce l'aggiornamento dei prezzi del punto vendita.
ControlloPuntoVendita	Permette al ResponsabilePuntoVendita di accedere al sistema per aggiornare i prezzi del punto vendita.
DisplayPrezzo	Gestisce la visualizzazione dei prezzi sugli scaffali.

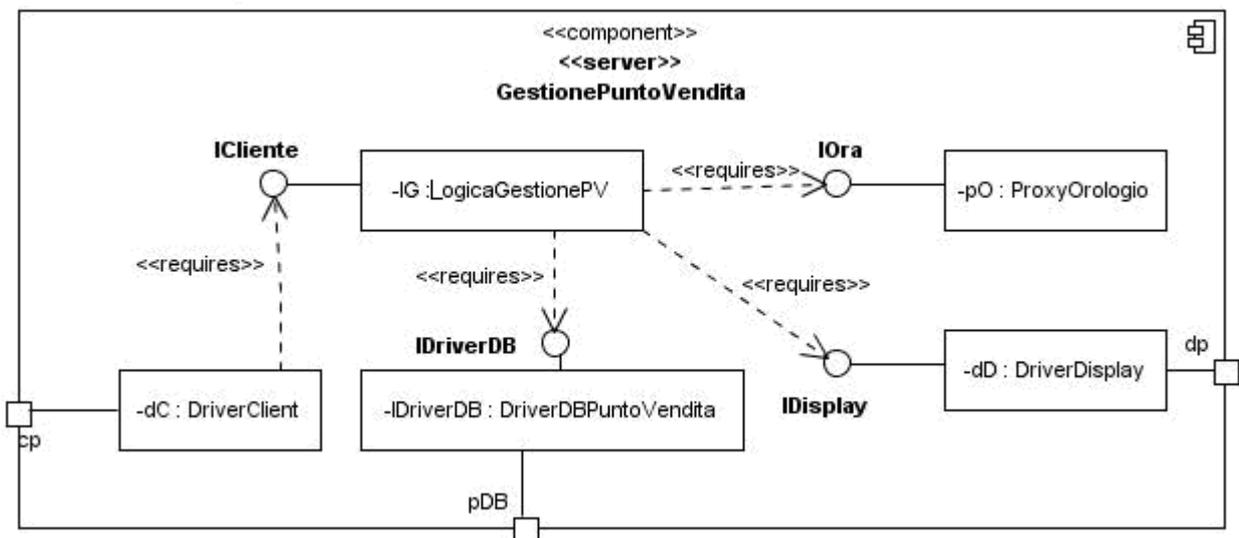
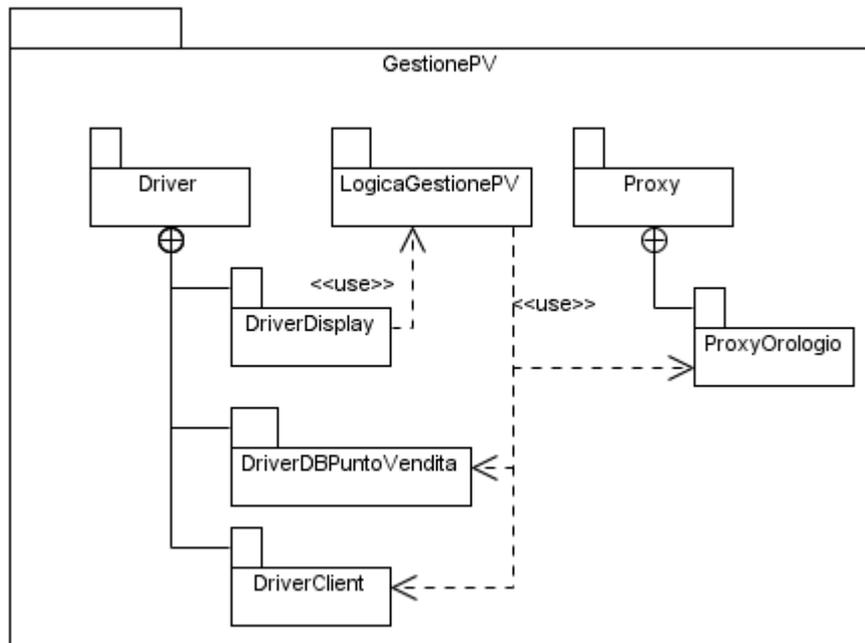


Figura 3 Componente GestionePuntoVendita

Definire la vista strutturale di decomposizione e d'uso che descrive le relazioni tra i package necessari a realizzare la componente GestionePuntoVendita

**Risposta.**



## 5. Verifica del software

### 19. Modifica dei prezzi.

Il seguente metodo è di fantasia e non realizza esattamente alcun caso d'uso (perché?). E' stato scritto per controllare la lista dei nuovi prezzi rispetto al vincolo di mantenere le variazioni entro il 20 per cento e aggiornare il data base. Le variabili cl e db denotano gli oggetti che gestiscono rispettivamente le interazioni col cliente e col data base.

```
public void controllaPrezzi(int [][] listaCodici) {
    for(int i=0; i < listaCodici.length; i++) {
        int codice = listaCodici[i][0];
//1
        int oldPrezzo = db.getPrezzo(codice);
//2
        int nuovoPrezzo = listaCodici[i][1];
//3
        if (nuovoPrezzo < oldPrezzo * 0.8){
            nuovoPrezzo = (int) (oldPrezzo * 0.8);
//4
        }
        if (nuovoPrezzo > oldPrezzo * 1.2) {
            nuovoPrezzo = (int) (oldPrezzo * 1.2);
//5
        }
        cl.aggiornato(codice, nuovoPrezzo);
//6
        db.setPrezzo(codice, nuovoPrezzo);
//7
    }
    return;
}
```

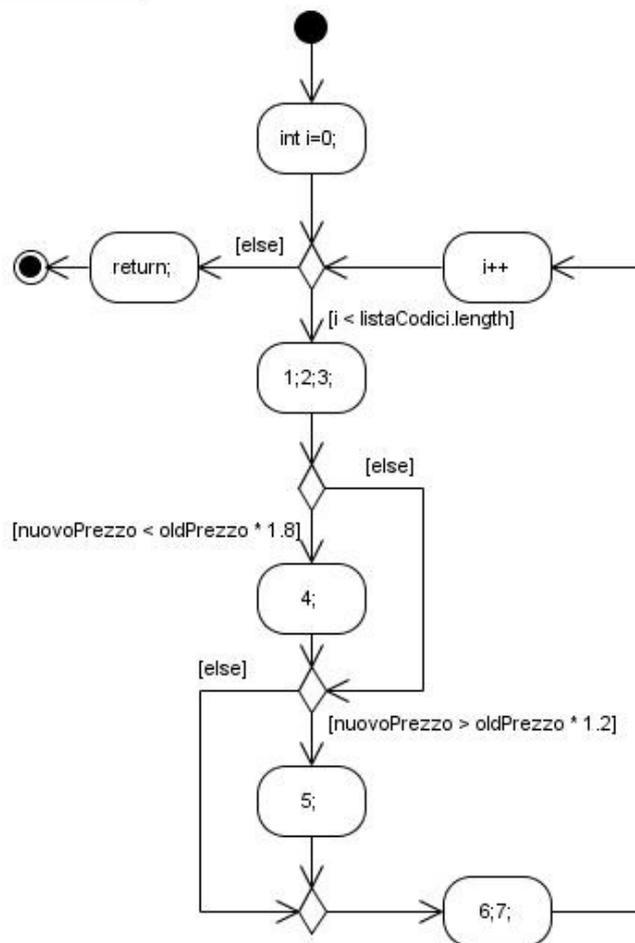
Dare un diagramma di attività con il grafo di flusso del metodo. Si usi la numerazione indicata per non ricopiare i comandi. Inoltre, per effettuare i test, si assuma che lo stub per getPrezzo sia definito in modo da restituire un valore pari a 10 volte il suo argomento. Gli argomenti dei metodi aggiornato e setPrezzo costituiscono l'output del metodo. Si utilizzi una tabella come la seguente per la risposta, con il numero di righe necessarie.

Input		Output
i	listaCodici[i]	

	<b>[0]</b>	<b>[1]</b>	<b>codice</b>	<b>nuovoPrezzo</b>

- Dare un (solo!) caso di test che soddisfi il criterio (a scatola aperta) delle decisioni.
- Dare un insieme minimale di casi di test soddisfi il criterio (a scatola aperta) di copertura dei comandi.
- Dare un insieme minimale di casi di test soddisfi il criterio (a scatola aperta) di copertura dei cammini, nel caso di 1-test dei cicli.

**Risposta:** Il grafo di flusso:



a) Un possibile caso di test è il seguente

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		
<b>0</b>	<b>1</b>	<b>11</b>	<b>1</b>	<b>11</b>
<b>1</b>	<b>2</b>	<b>15</b>	<b>2</b>	<b>16</b>
<b>2</b>	<b>3</b>	<b>40</b>	<b>3</b>	<b>36</b>

b) Basta una lista con due prezzi, fuori dall'intervallo permesso, dalle due parti:

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		
1	2	15	2	16
2	3	40	3	36

c) Servono tre liste di un solo elemento, per fare tre cicli di un solo passo, in ciascuno dei quali si percorre uno dei tre cammini possibili nel corpo del ciclo:

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		
0	1	11	1	11

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		
0	2	15	2	16

Input			Output	
i	listaCodici[i]		codice	nuovoPrezzo
	[0]	[1]		
0	3	40	3	36

## 20. Cassa per gli acquisti.

Per calcolare il prezzo totale di una spesa è stato scritto un metodo che riceve in input un array con i codici degli articoli acquistati e restituisce il totale della spesa. In particolare, il metodo tiene conto delle offerte 3x2 (per ogni 3 articoli in offerta comperati, se ne pagano solo 2).

- Fornire un caso di prova per verificare se il metodo si comporta correttamente per quanto riguarda la gestione delle offerte 3x2. Per la progettazione del caso di prova si usi il criterio funzionale basato sulla partizione dei dati in ingresso. Si assuma che i codici siano interi, e che, nell'ambiente di test, tutti i prodotti in vendita siano in offerta e costino 10 volte il loro codice.
- Verificare se l'esecuzione del codice seguente per il caso di prova dato in a) evidenzia un malfunzionamento. La variabile db denota l'oggetto che gestisce le interazioni col data base.

```

public static int calcola (int [] articoliComprati) {
    int totale = 0;
    for(int i=0; i < articoliComprati.length; i++) {
        int articolo = articoliComprati[i];
        if(articolo != -1 ){
            int prezzo = db.getPrezzo(articolo);
            if(db.offerta(articolo)) {
                int count = 1;
                for(int j=i+1; j<articoliComprati.length; j++) {
                    if(articolo == articoliComprati[j]) {
                        count++;
                        articoliComprati[j]=-1;
                    }
                }
                prezzo = prezzo*2*(count/3);
            }
            totale = totale + prezzo;
        }
    }
    return totale;
}

```

**Risposta:**

a)

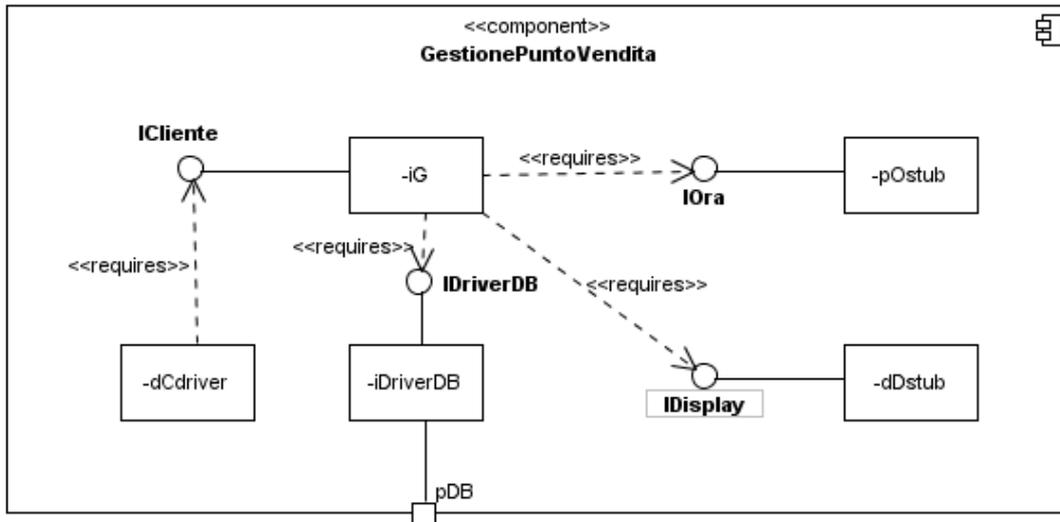
Input	Output
[1,2,3,2,3,3]	110

b) L'output del metodo è invece 60 (=  $10*2*(1/3) + 20*2*(2/3) + 30*2*(3/3) = 0+0+60$  ).

**21. Modifica dei prezzi (Ambiente).**

Descrivere, con un diagramma di struttura composita, l'ambiente di verifica (stub e driver) di LogicaGestionePV in Figura 3. (Si assuma di avere già testato il database, e quindi poterlo utilizzare per il test).

**Risposta.**



(Nota: i tipi si desumono dalla Figura 3)

1. (Verifica del software) Si consideri la modifica dei prezzi

Il metodo **int getPrezzo(int codice)** restituisce il prezzo corrente di un prodotto.

Il metodo **int controllaPrezzi(int codice, int prezzo)** controlla i nuovi prezzi rispetto al vincolo di mantenere le variazioni entro il 20 per cento. Riceve in input il codice di un prodotto e il nuovo prezzo e restituisce il prezzo da inserire nel database, avendolo eventualmente adeguato per rispettare il vincolo sul 20%.

Si assuma che il metodo getPrezzo dia i seguenti risultati:

getPrezzo(1)= 20	getPrezzo(4)= 20
getPrezzo(2)= 35	getPrezzo(5)= 10
getPrezzo(3)= 15	getPrezzo(6)= 7

Usando criteri a scatola chiusa, definire una batteria di test per il metodo controllaPrezzi. Motivare la scelta effettuata.

**Risposta:**

Una possibile batteria di test è la seguente:

1. aggiornamento valido
2. eccesso di ribasso
3. ribasso limite
4. eccesso di rialzo
5. rialzo limite

Input		Output
codice	prezzo	
1	21	21
2	15	28
3	12	12
4	25	24
5	12	12