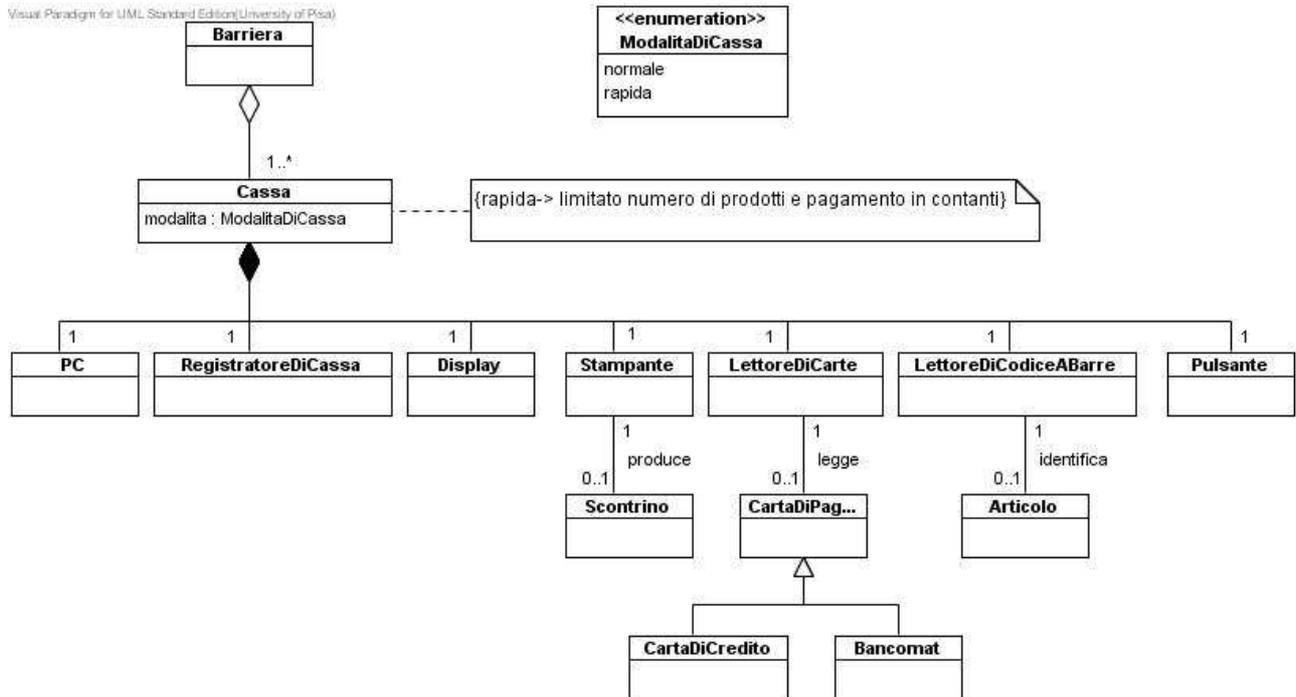


La prova si svolge a libri chiusi (non è permessa la consultazione di materiale didattico).  
Traccia delle soluzioni.

Si consideri il caso di studio 2, Grande distribuzione, e in particolare la gestione della cassa per gli acquisti.

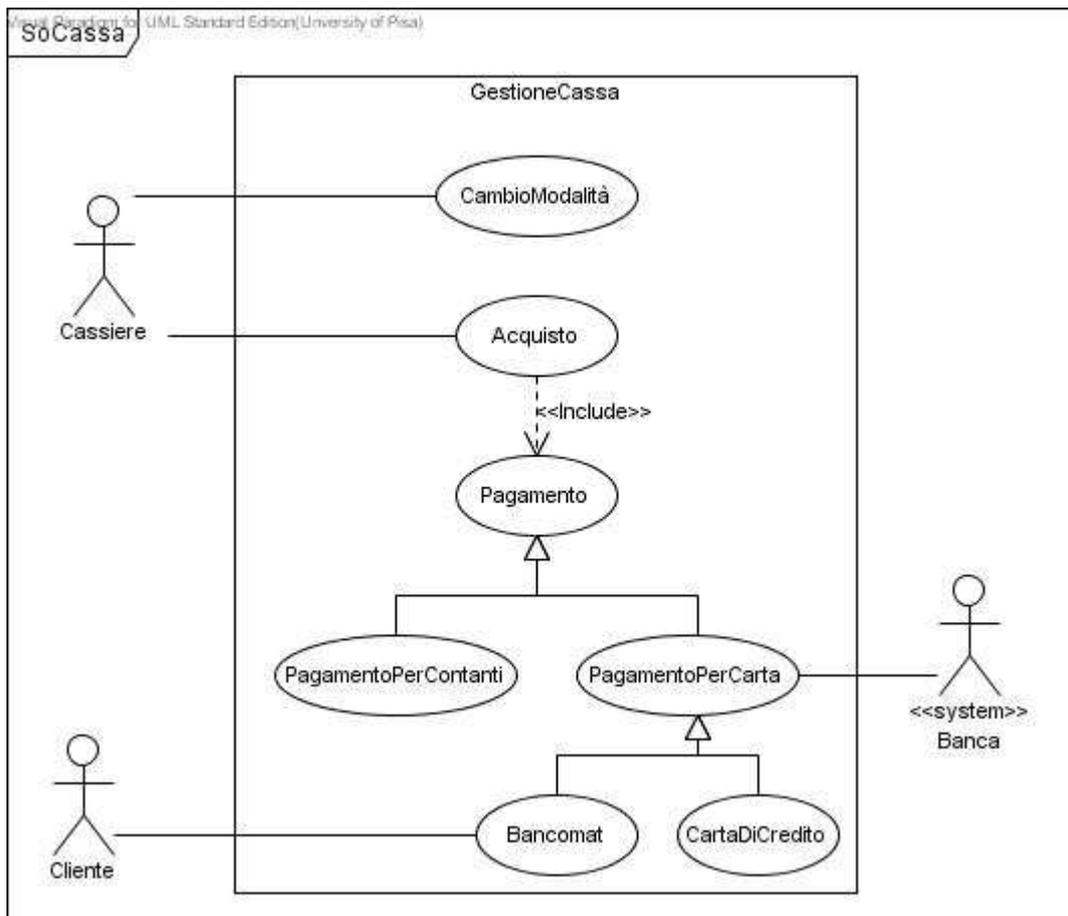
**Domanda 1.** (Analisi del dominio). Dare un diagramma delle classi, considerando come classi, attributi o costanti tutti e soli i termini seguenti: articolo, bancomat, barriera, carta di credito, carta di pagamento, cassa, display, lettore di carte, lettore di codice a barre, modalità, normale (modalità di cassa), PC, pulsante di commutazione modalità di cassa, rapida (modalità di cassa), registratore di cassa, scontrino, stampante.

**Risposta.**



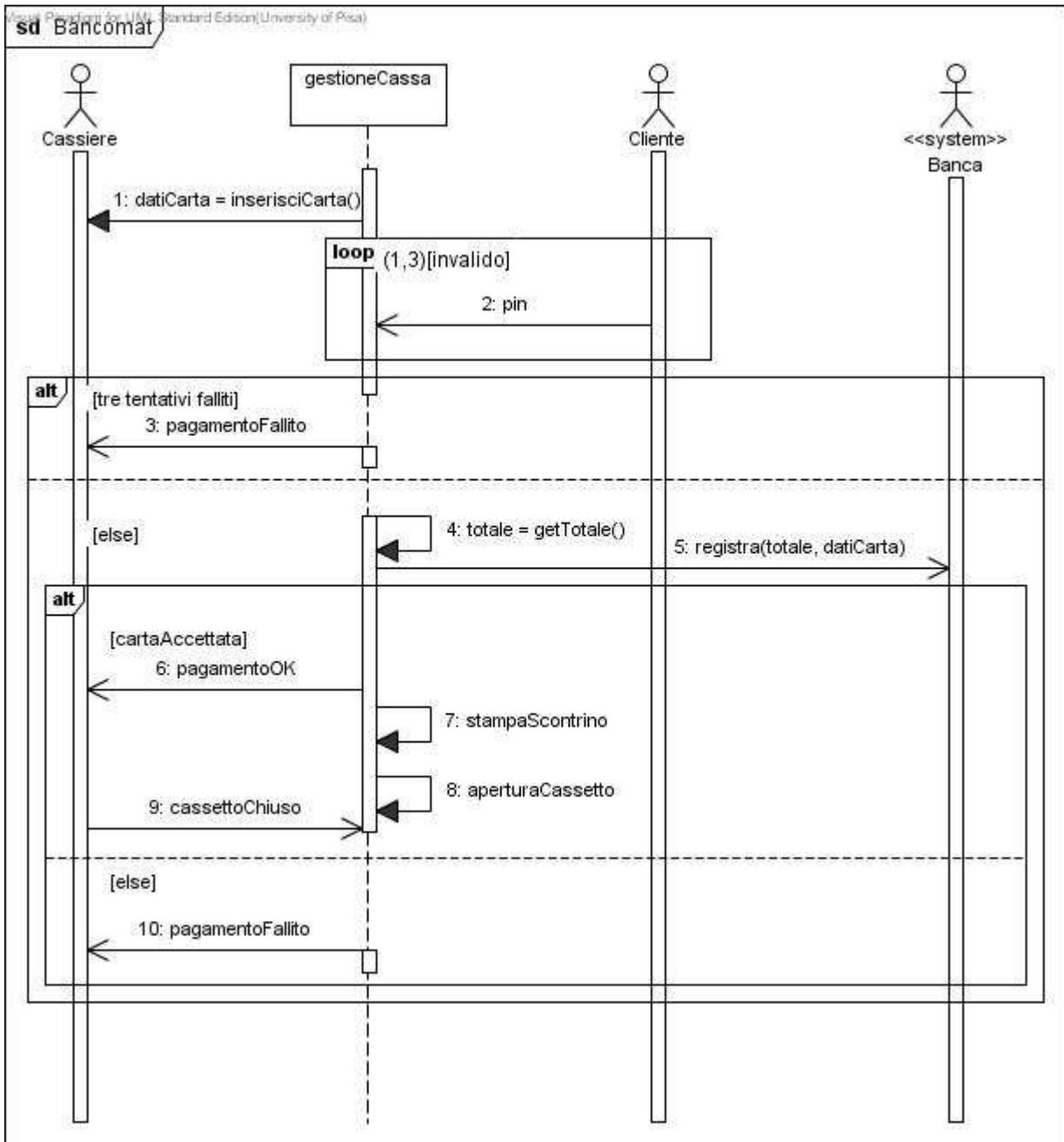
\*\*\*\*\*

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:



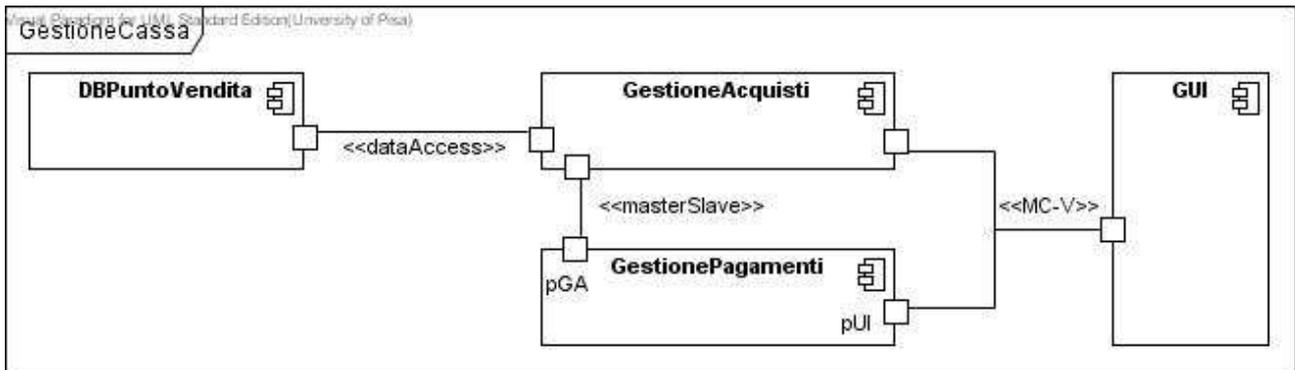
**Domanda 2.** (Analisi dei requisiti). Dare, con un diagramma di sequenza, la specifica del caso d'uso pagamento con bancomat. Si assumano le seguenti operazioni interne di GestioneCassa: get-Totale (per ottenere il totale da pagare), stampaScontrino (per la stampa delle ricevute) e apriCassetto (per 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.**



\*\*\*\*\*

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



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.

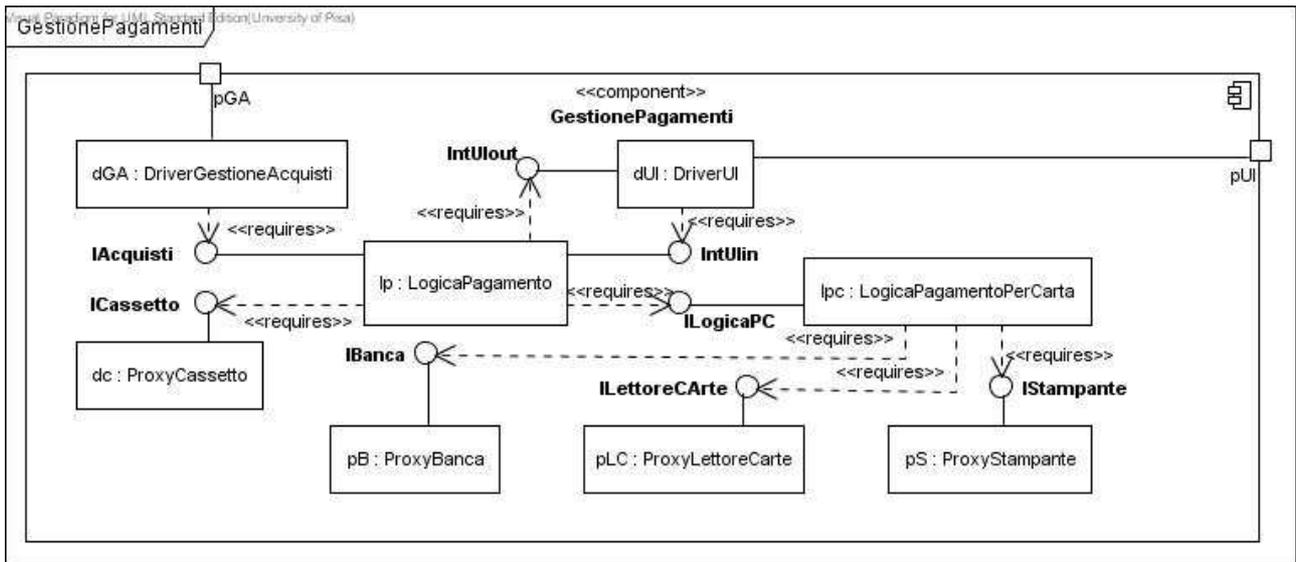
Le componenti hanno le seguenti responsabilità:

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.

**Domanda 3.** (Progettazione di dettaglio) Dare un diagramma di struttura composita della componente GestionePagamenti. Si distingue tra la logica di pagamento e quella di pagamento con carte.

**Risposta.**

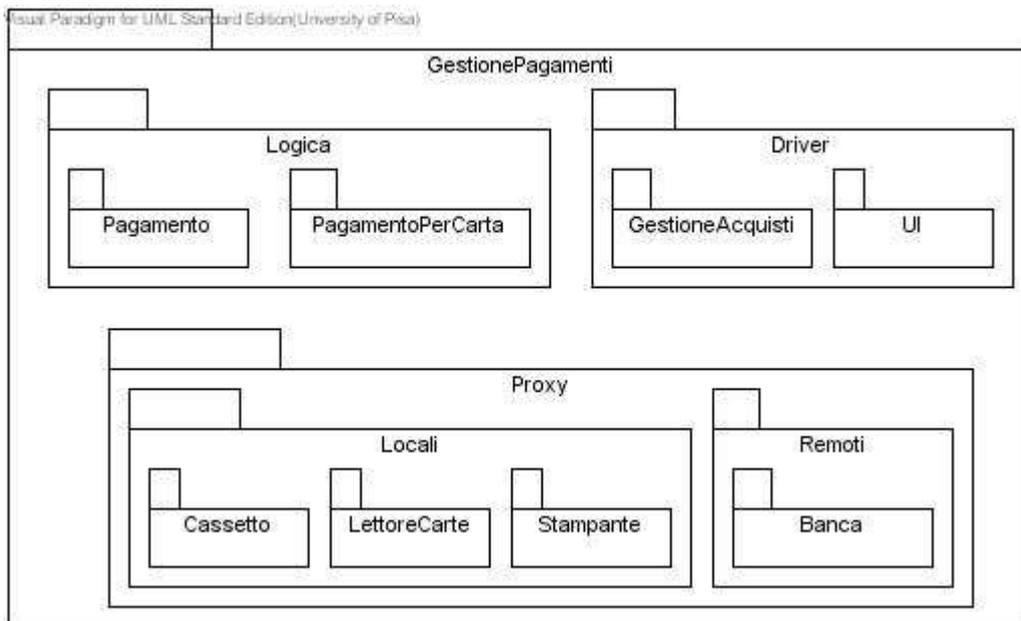


\*\*\*\*\*

**Domanda 4.** (Architettura) Fornire una vista strutturale di decomposizione del modulo GestionePagamenti, in base alle seguenti regole:

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

**Risposta.**



\*\*\*\*\*

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).

**Domanda 5.** (Verifica del software)

- a) 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.
- b) 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 (=  $10*2*(1/3) + 20*2*(2/3) + 30*2(3/3) = 0+0+60$ ).