

5. Elaborazione – Iterazione 4

5.1 Introduzione

I casi d'uso che verranno trattati nella quarta iterazione sono:

- Caso d'uso UC7: Aggiunti sconto, con questo caso d'uso sarà possibile associare sconti precedentemente creati nel sistema tramite il caso d'uso UC8 (Gestisci politiche di sconto).
- Caso d'uso UC9: Effettua pagamento, grazie a questo caso d'uso sarà possibile effettuare un pagamento con l'utilizzo di diversi metodi elettronici.

Entrambi i casi d'uso sono delle sottofunzioni del caso d'uso UC1 (Gestisci Comanda).

5.2 Aggiornamento del modello dei casi d'uso

Nella fase di ideazione i casi d'uso UC7 e UC9 sono stati descritti in modo breve, dato che in questa iterazione saranno analizzati e implementati, vengono quindi descritti in modo dettagliato dopo l'incontro con il committente.

UC7: Aggiungi sconto

Nome del caso d'uso	UC7: Aggiungi sconto
Portata	Pizzeria Panucci
Livello	Sottofunzione
Attore primario	Cliente
Parti interessate e interessi	<ul style="list-style-type: none">- Cliente: vuole verificare in modo semplice se è possibile applicare uno sconto sulla sua comanda- Titolare: vuole che gli sconti siano applicati in modo corretto secondo le politiche interne della Pizzeria
Pre-condizioni	C'è una comanda in corso
Garanzia di successo	
Scenario principale di successo	<ol style="list-style-type: none">1. Il Cliente conferma di aver finito con l'inserimento di pizze nella comanda2. Il Sistema verifica la presenza di uno sconto applicabile3. Il Sistema calcola lo sconto e lo applica sull'importo della comanda
Estensioni	2a. Il sistema non trova sconti applicabili <ol style="list-style-type: none">1. Il sistema non applica sconti
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Ogni volta che viene eseguita una comanda
Varie	

UC9: Effettua pagamento

Nome del caso d'uso	UC9: Effettua pagamento
Portata	Pizzeria Panucci
Livello	Sottofunzione
Attore primario	Cliente
Parti interessate e interessi	<ul style="list-style-type: none"> - Cliente: vuole pagare per via elettronica - Titolare: vuole ricevere i pagamenti per via elettronica
Pre-condizioni	C'è una comanda in corso
Garanzia di successo	Il pagamento è andato a buon fine
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il Cliente conferma la comanda 2. Il Sistema chiede la tipologia di pagamento 3. Il Cliente seleziona la tipologia di pagamento "con carta" 4. Il Cliente inserisce i dati della propria carta 5. Il Sistema verifica i dati immessi dal cliente ed effettua il pagamento
Estensioni	3a. La tipologia di pagamento selezionata è "paypal" <ol style="list-style-type: none"> 1. Il Cliente inserisce i suoi dati 2. Il sistema verifica i dati ed effettua il pagamento
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizioni	Ogni volta che viene eseguita una comanda
Varie	

5.2.1 Aggiornamento del glossario

Per questa iterazione si è ritenuto necessario un aggiornamento del glossario inserendo i parametri richiesti dai diversi metodi di pagamento e della classe concettuale sconto.

Glossario

1. **Sconto:** sconto applicato dal sistema sul prezzo della comanda.
2. **Comanda:** termine riferito all'ordine effettuato dal cliente. Questo implica che il cliente ha già pagato l'ammontare complessivo.
3. **Ingrediente:** termine che si riferisce ai componenti della pizza.
4. **Paypal:** metodo di pagamento elettronico che richiede Username e Password.
5. **Carta di Credito:** metodo di pagamento elettronico che richiede Nome del titolare, numero della carta, cvv e scadenza della carta.
6. **Carta di Debito:** metodo di pagamento elettronico che richiede Nome del titolare, numero della carta, cvv e scadenza della carta.

5.3 Analisi Orientata agli Oggetti

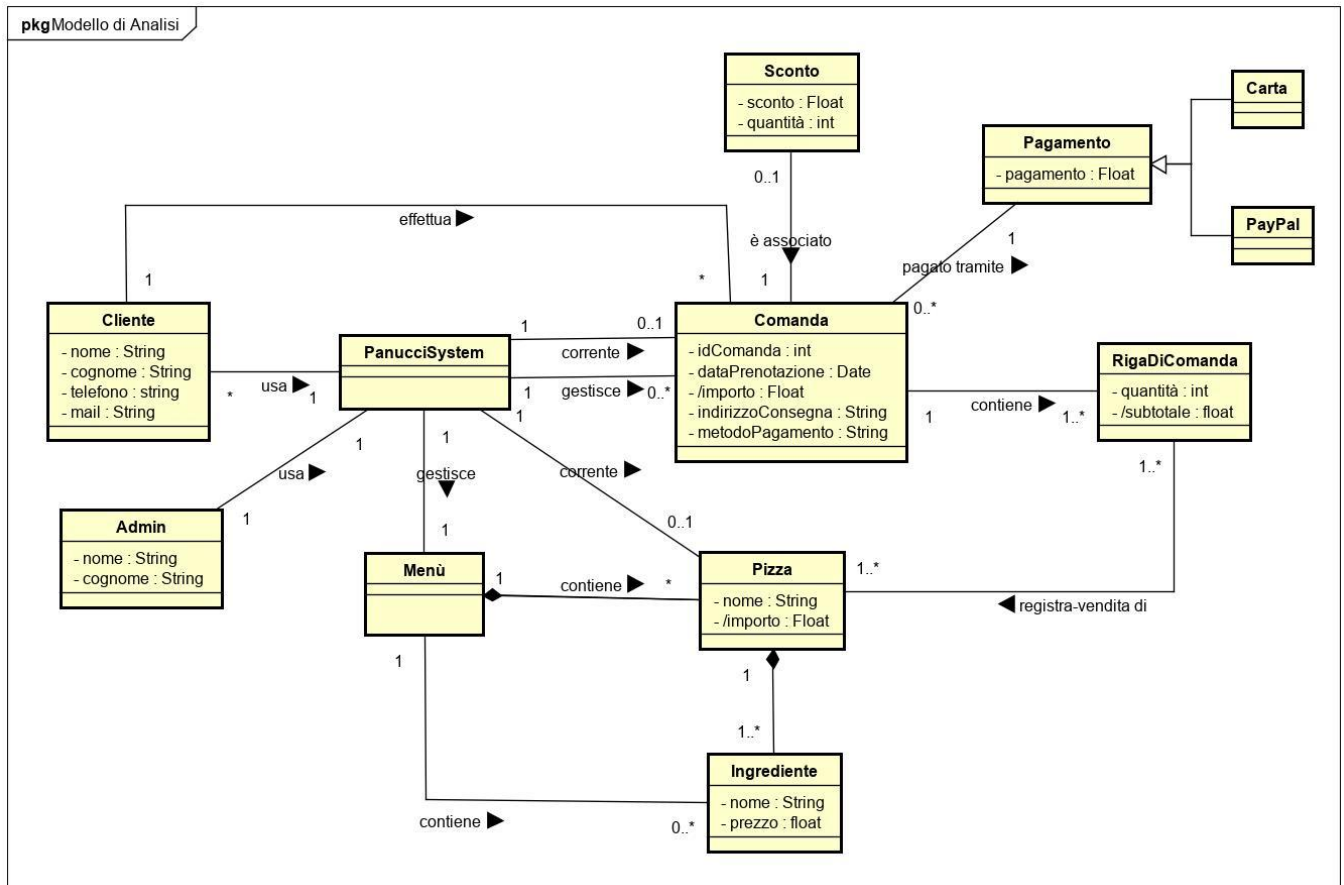
Come per la precedente iterazione saranno considerati in questa fase il modello di dominio, il diagramma di sequenza di sistema e i rispettivi contratti delle operazioni.

5.3.1 Modello di dominio

Analizzando il caso d'uso Effettua Pagamento si è ritenuto necessario aggiungere una gerarchia di classi per la rappresentazione dei vari metodi di pagamento.

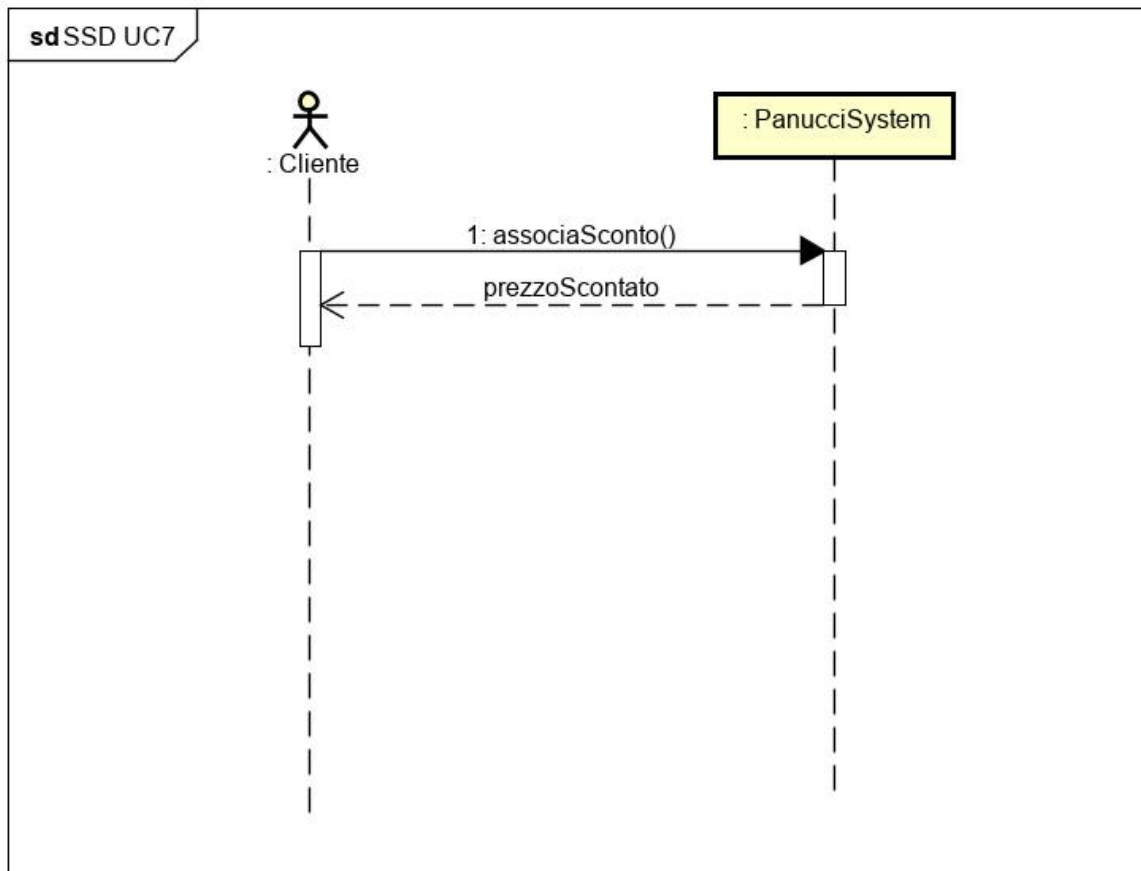
Inoltre viene aggiunta la classe concettuale Sconto che contiene le informazioni sul prezzo di sconto applicato e sulla quantità.

Viene dunque riportato il modello di dominio aggiornato:

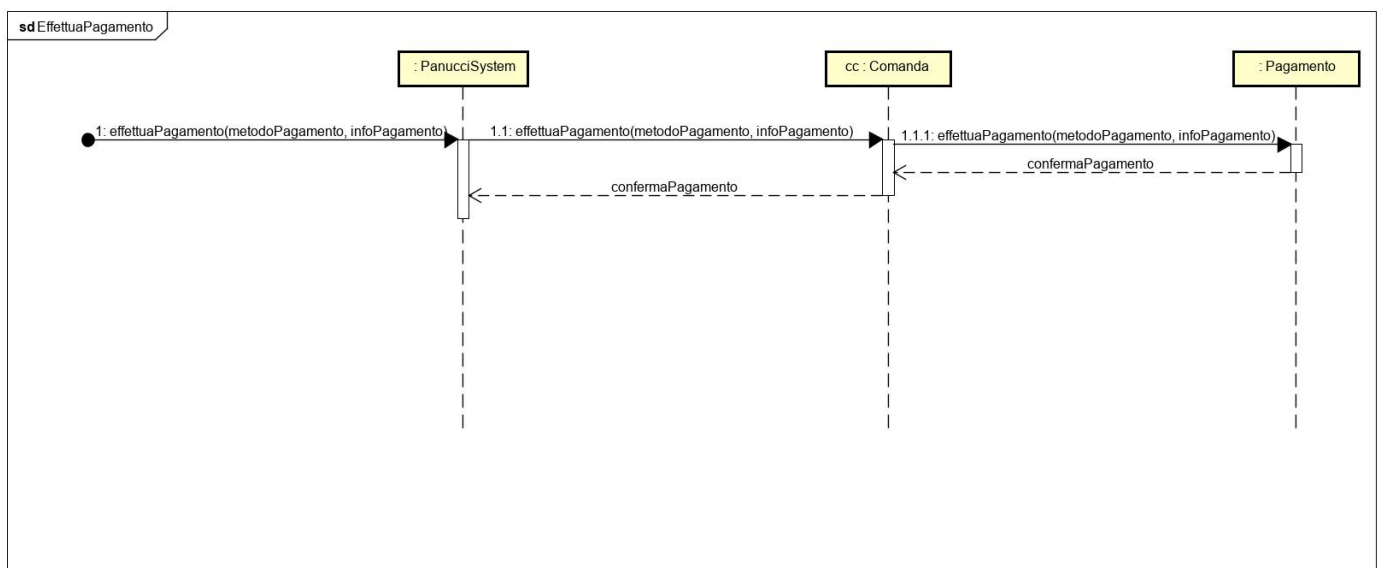


5.3.2 Diagramma di sequenza di sistema

Lo scenario principale di successo del caso d'uso UC7 è illustrato nel sottostante diagramma di sequenza di sistema:



Lo scenario principale di successo del caso d'uso UC9 viene riportato con lo schema seguente:



5.3.3 Contratto delle operazioni

5.3.3.1 associaSconto

L'operazione di sistema associaSconto avrà il compito di associare uno sconto alla comanda in corso ed aggiornare il prezzo relativo.

Operazione	associaSconto(sconto)
Riferimenti	Caso d'uso: UC7 Aggiungi Sconto
Pre-condizioni	- c'è una comanda in corso
Post-condizioni	- l'attributo prezzoScontato della comanda in corso è stato aggiornato.

5.3.3.2 effettuaPagamento

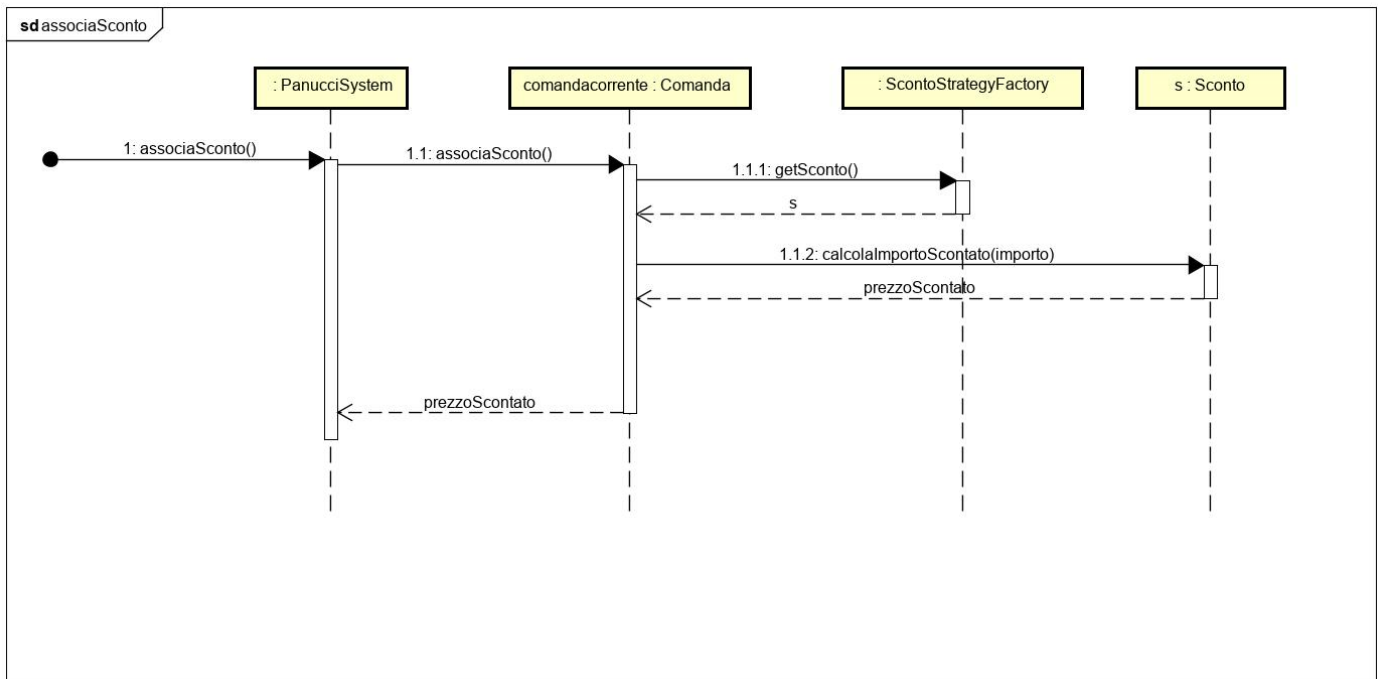
L'operazione di sistema effettuaPagamento si occuperà di effettuare il pagamento.

Operazione	effettuaPagamento(metodoPagamento, descrizionePagamento)
Riferimenti	Caso d'uso: UC9 Effettua pagamento
Pre-condizioni	- c'è una comanda in corso
Post-condizioni	- è stata creata un'istanza della classe concreta che implementa l'interfaccia Pagamento - è stato chiamato il metodo paga() che inoltra la chiamata al sistema esterno che si occupa di gestire il pagamento - è stata restituita la conferma che il pagamento si è concluso a buon fine.

5.4 Progettazione

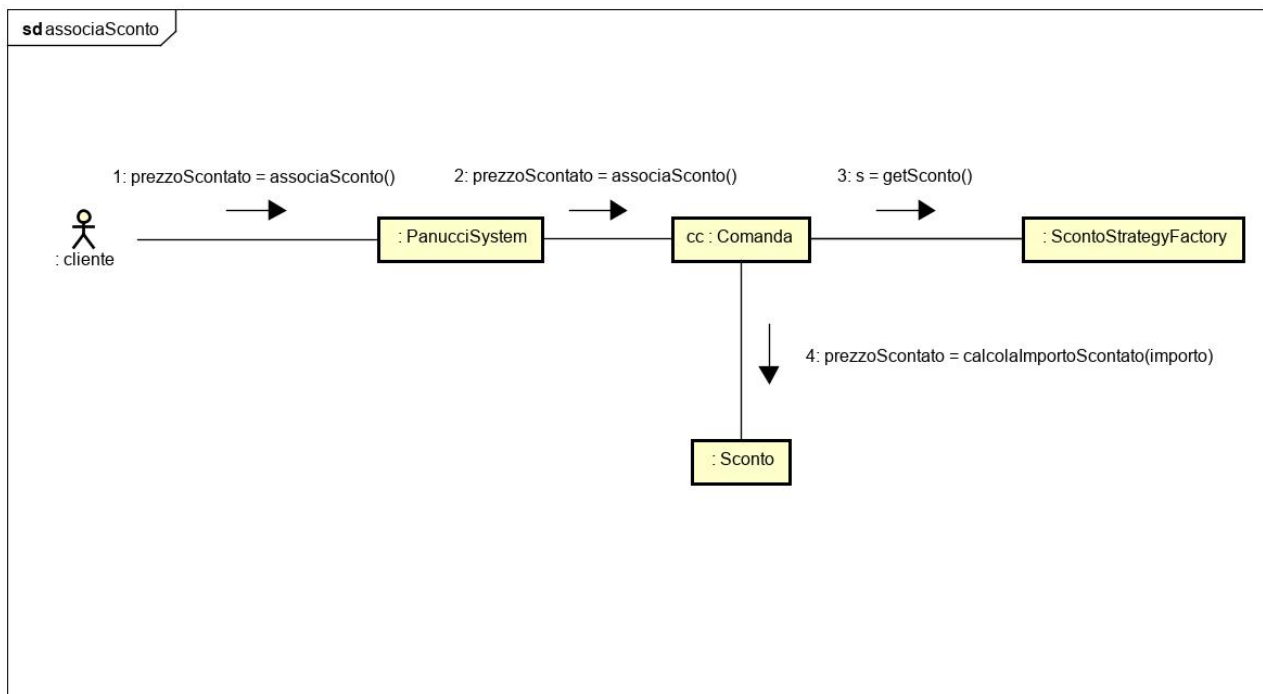
Si eseguono adesso i diagrammi di sequenza relativi ai casi d'uso UC7 e UC9.

5.4.1. associaSconto



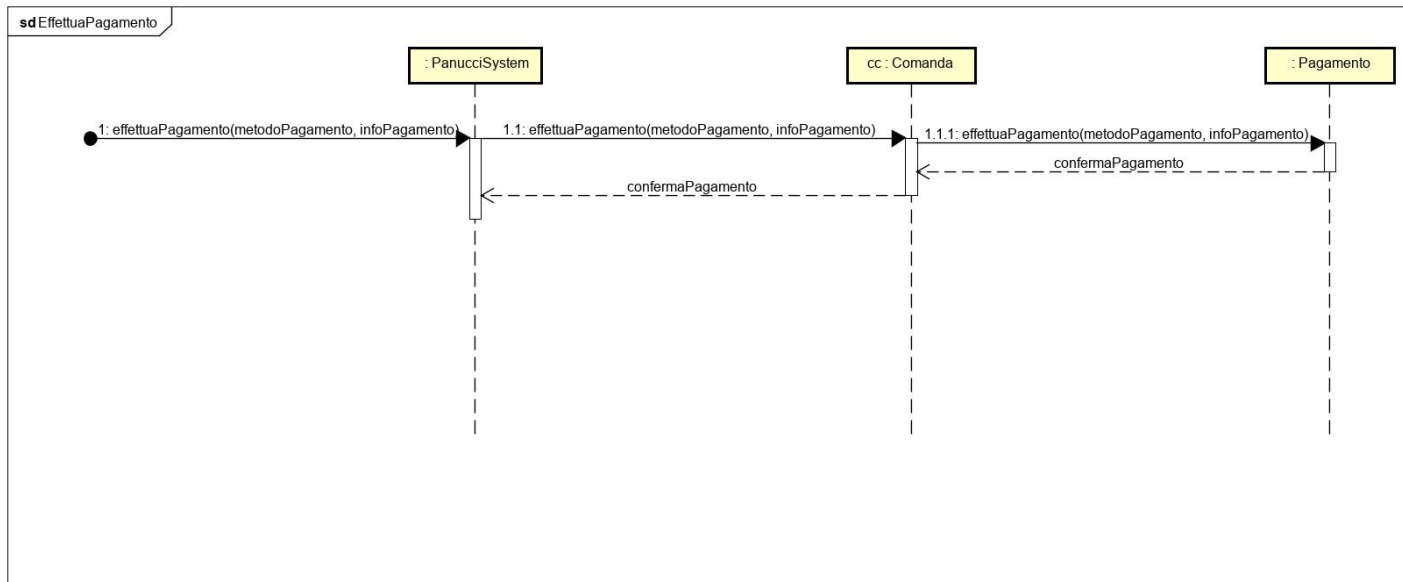
Si noti che è stata utilizzata una factory che avrà il compito di creare lo sconto opportuno secondo un file di configurazione esterno, nel progetto è stata simulato la creazione di uno scontoAssoluto.

In questa occasione viene riportato anche il diagramma di comunicazione.

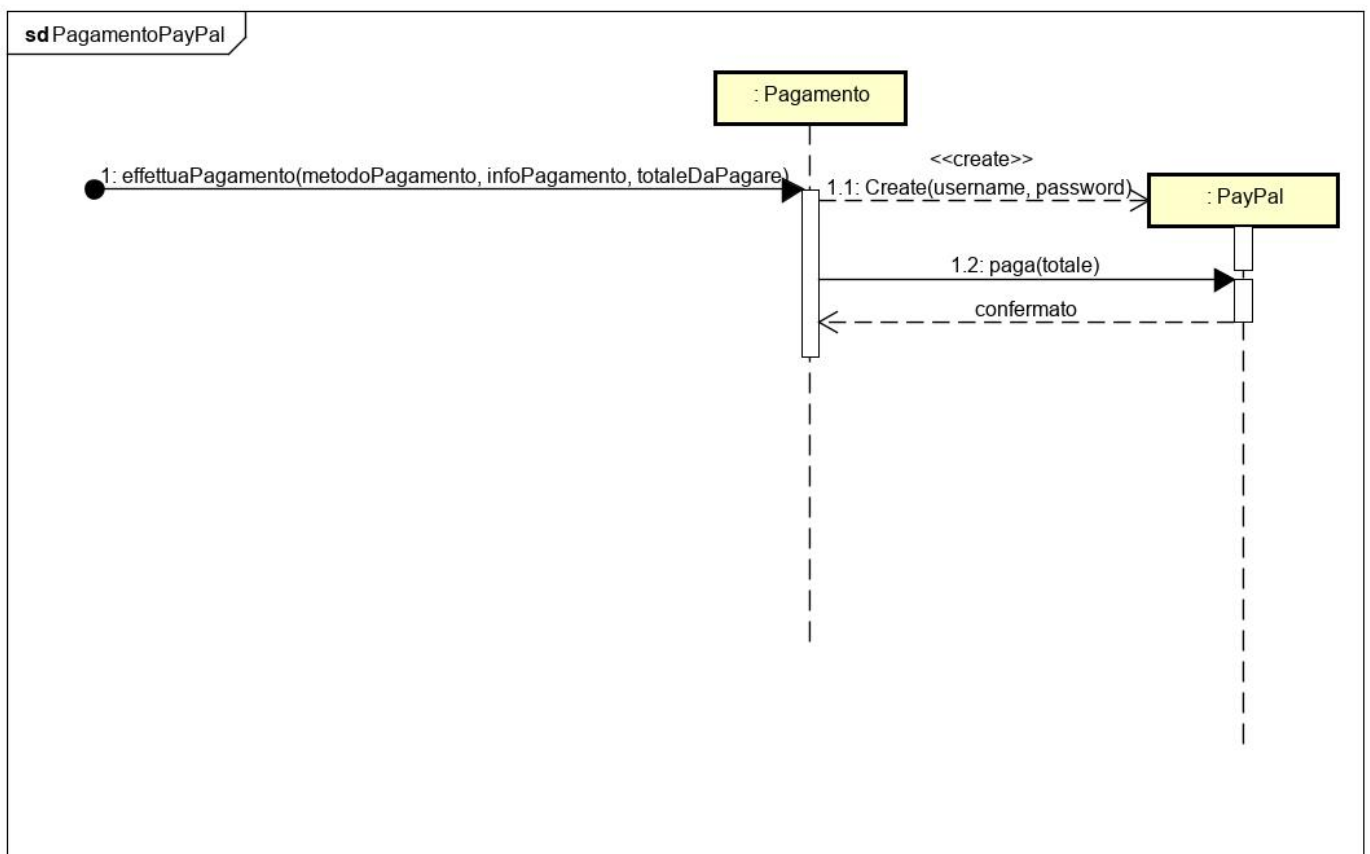


5.4.2. effettuaPagamento (metodoPagamento, infoPagamento)

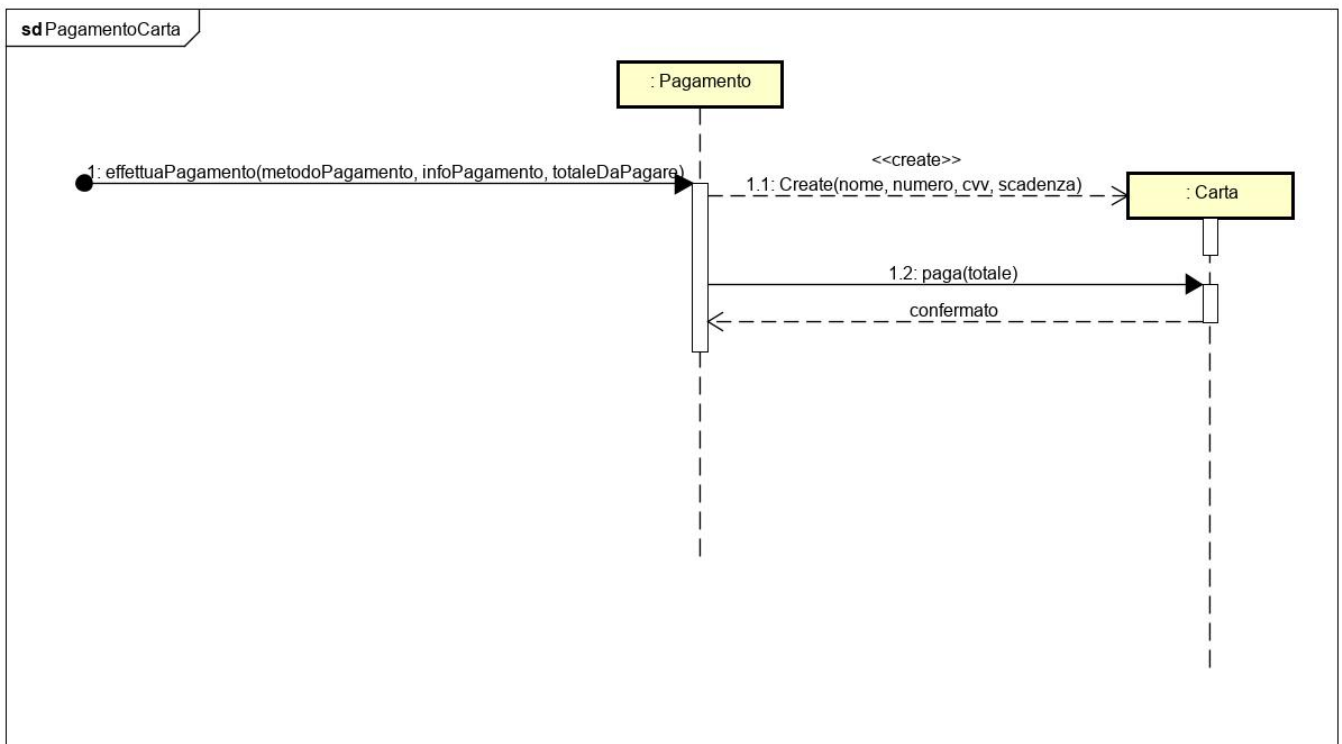
Dato che gli scenari alternativi prevedono l'utilizzo di differenti metodi di pagamenti, i quali vengono realizzati attraverso l'ausilio del pattern factory method e strategy per la gestione dell'algoritmo da utilizzare per il pagamento, affinché si ottenga una maggiore flessibilità nell'introduzione di nuovi algoritmi in futuro. Tenuto conto di queste considerazioni, vengono di seguito riportati i diagrammi per tutti gli scenari alternativi possibili.



5.4.2.1. Effettua Pagamento PayPal

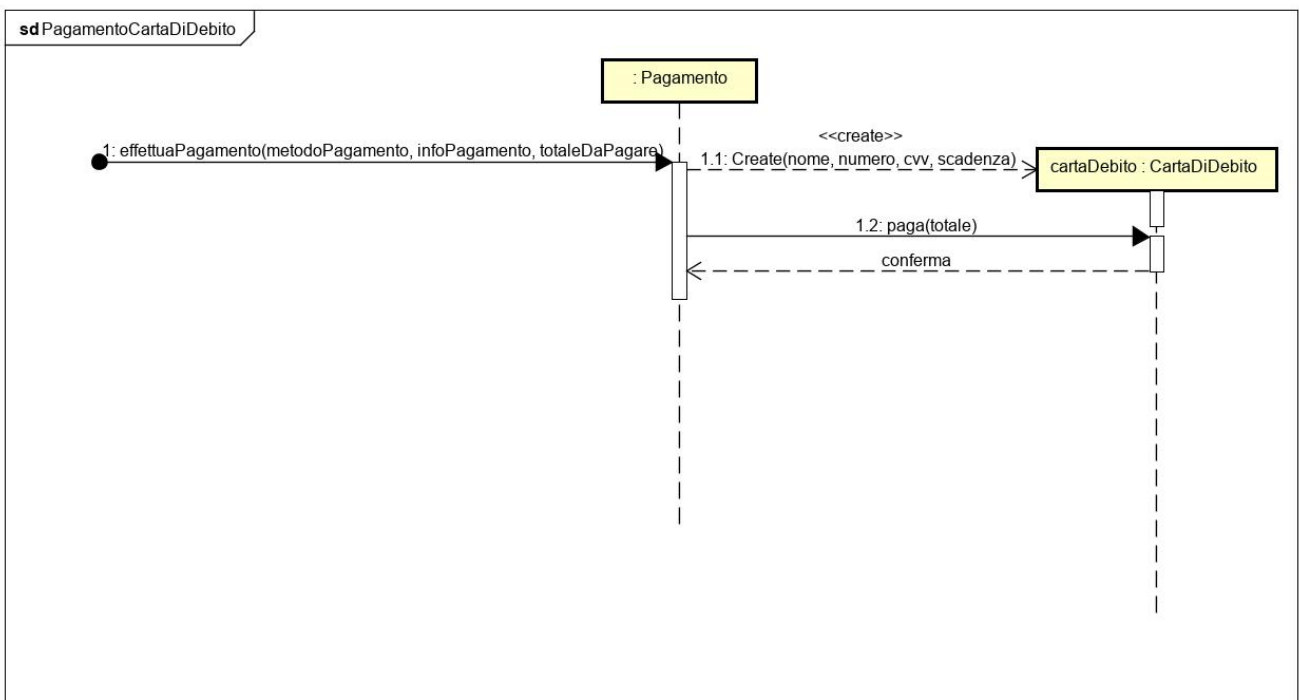


5.4.2.2. Effettua Pagamento Carta

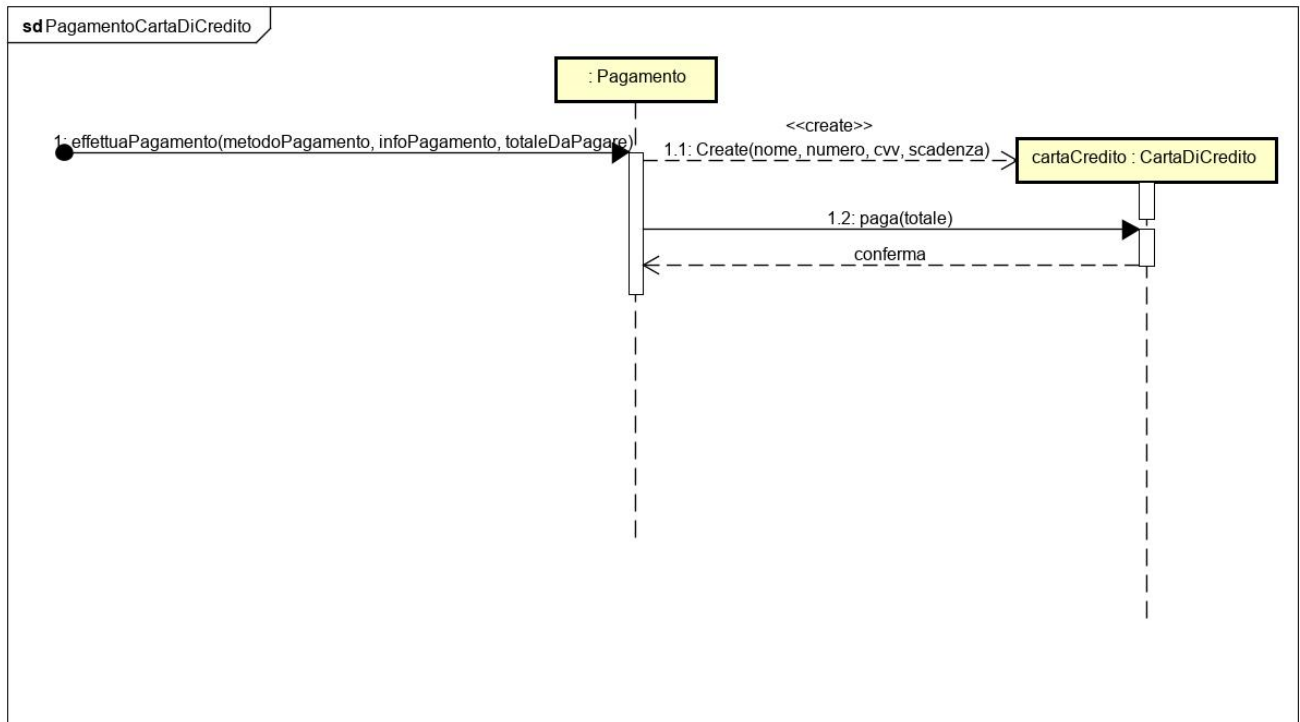


Visto che è stato utilizzato il polimorfismo in Carta la quale è una classe astratta con le sottoclassi concrete CartaDiDebito e CartaDiCredito sono riportati i diagrammi per ogni caso.

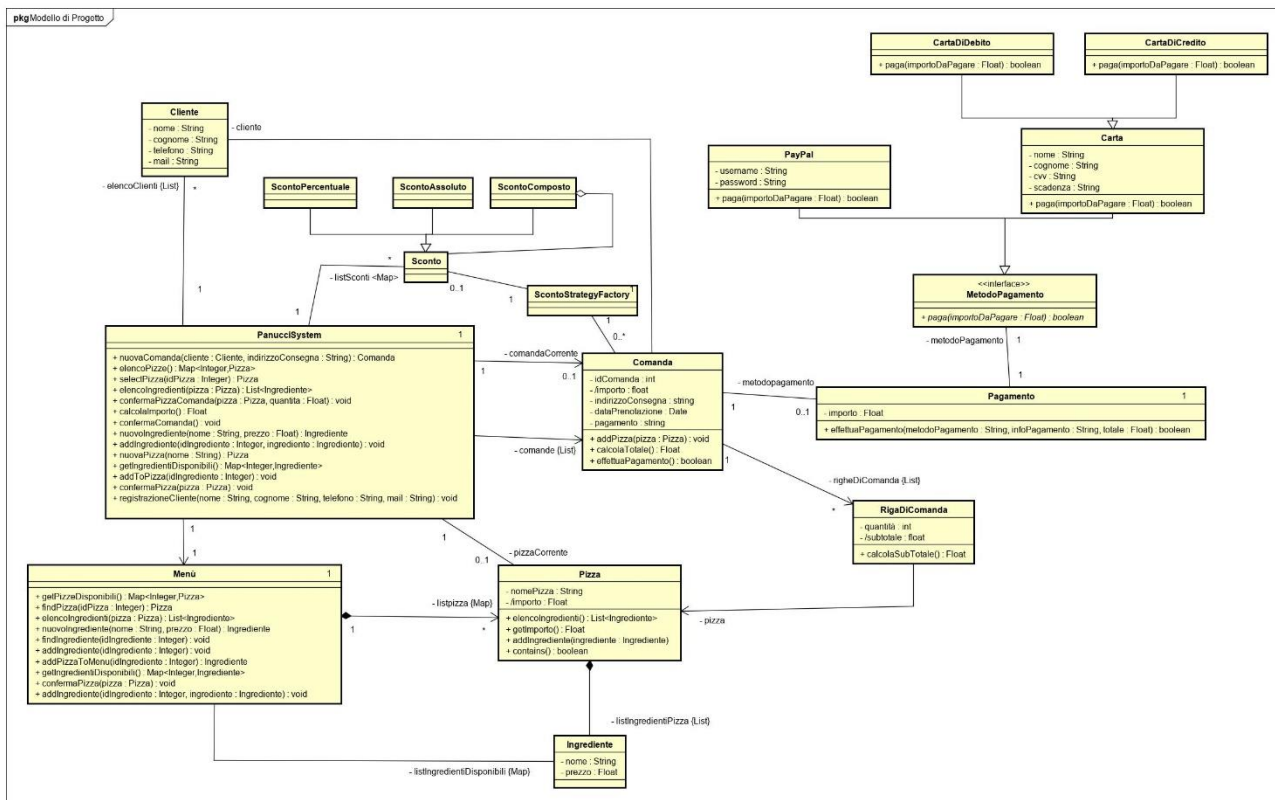
5.4.2.2.1 Effettua Pagamento Carta, caso CartaDiDebito



5.4.2.2.1 Effettua Pagamento Carta, caso CartaDiCredito



5.5. Diagramma delle classi di progetto



Le classi che sono state introdotte in questa iterazioni sono:

- Sconto
- ScontoPercentuale
- ScontoAssoluto
- ScontoComposto
- ScontoStrategyFactory (Singleton)
- Pagamento (Singleton)
- MetodoPagamento
- PayPal
- Carta
- CartaDiDebito
- CartaDiCredito

È stato utilizzato il pattern Composite per ottenere uno sconto che sia anche cumulabile, inoltre è stata introdotta la classe ScontoStrategyFactory che viene usata per togliere la responsabilità della creazione dello sconto alla classe Comanda, sarà la factory ad occuparsene, Comanda potrà ricevere lo sconto opportuno dalla factory, la quale saprà che sconto applicare in base ad un file di configurazione, un file in un Database o altro, nell'implementazione è stato ipotizzata la creazione di uno ScontoAssoluto per semplicità.

è stato utilizzato il pattern Strategy in MetodoDiPagamento, in questo modo è possibile aggiungere in modo semplice un nuovo metodo di pagamento semplicemente creando una nuova classe che implementa MetodoDiPagamento. Nel caso in esame, Strategy è stato utilizzato in combinazione con Factory Method, in modo da spostare la responsabilità della creazione della classe alla factory chiamata in questa occasione Pagamento, la classe corretta sarà istanziata a seconda dei parametri forniti in input in maniera trasparente all'utente.