

# Iterazione 5

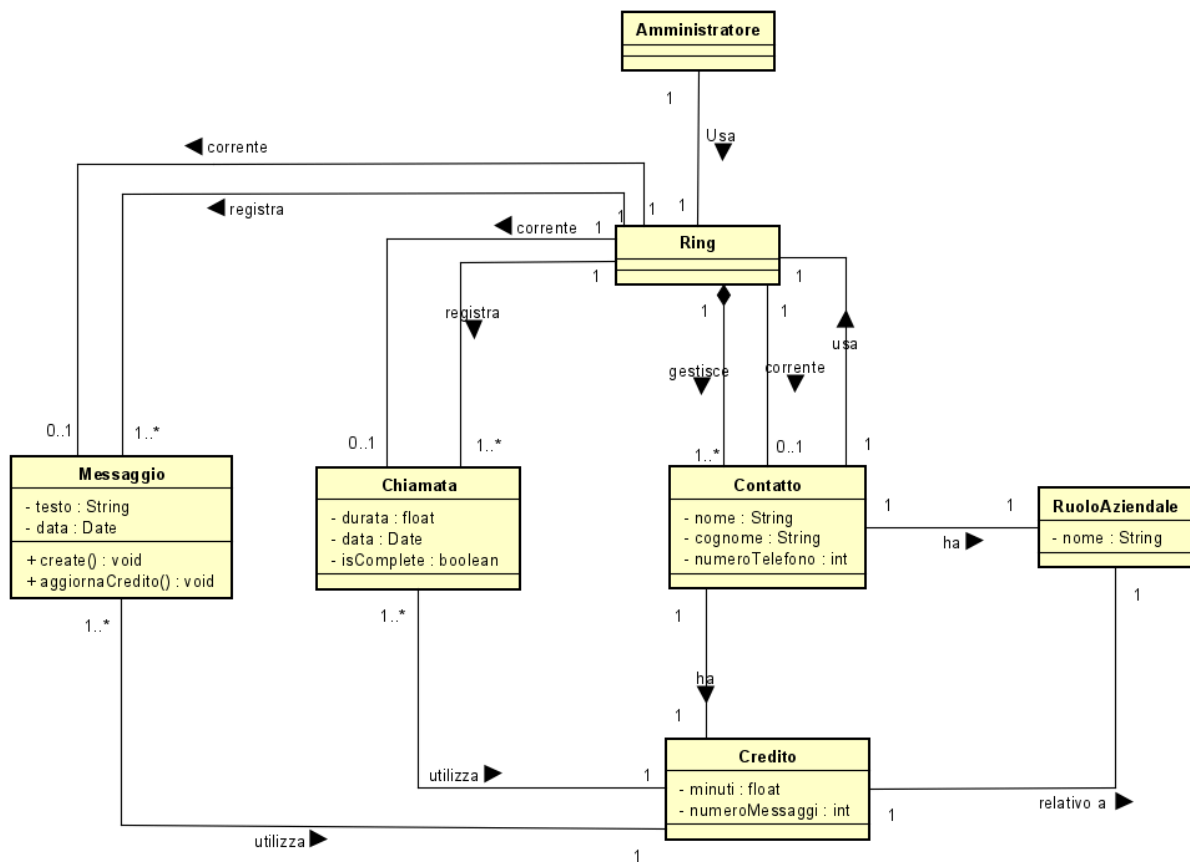
## 1. Introduzione

Nella iterazione 5 ci concentriamo sull'implementazione caso d'uso UC6->UC11, il modello di dominio, diagrammi di sequenza di sistema SSD, contratti delle operazioni, diagramma delle classi, la stesura dei diagrammi di sequenza ed infine sul miglioramento della gestione del credito tramite l'utilizzo del pattern Strategy.

## 2. Analisi Orientata agli oggetti

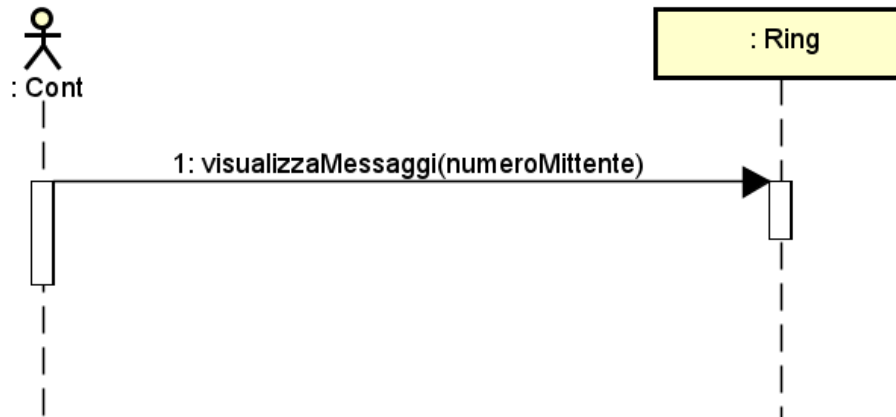
### 2.1 Modello di Dominio

Il modello di dominio rimane uguale all'iterazione precedente.

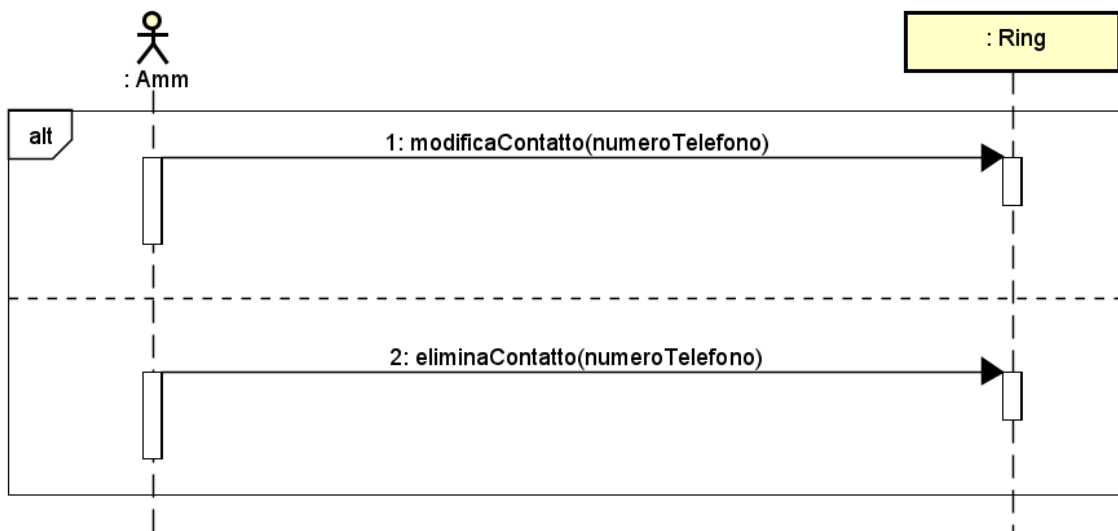


## 2.2 Diagramma di sequenza di sistema

- UC5:



- UC6:



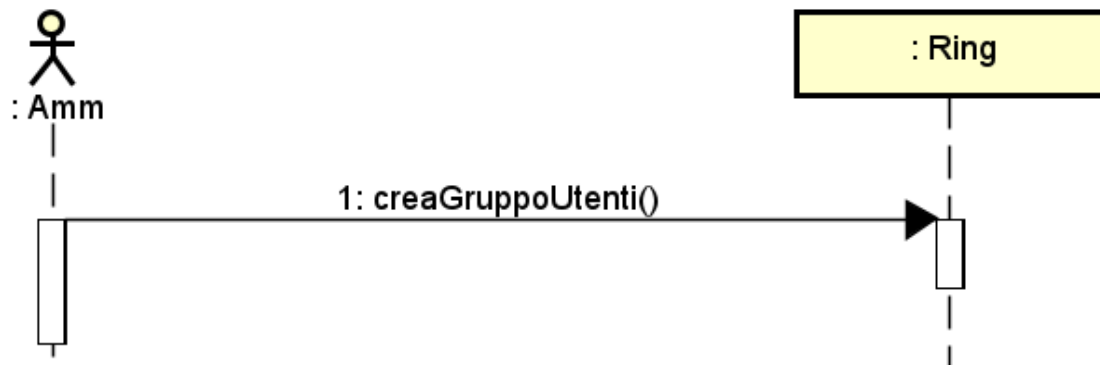
- UC7:



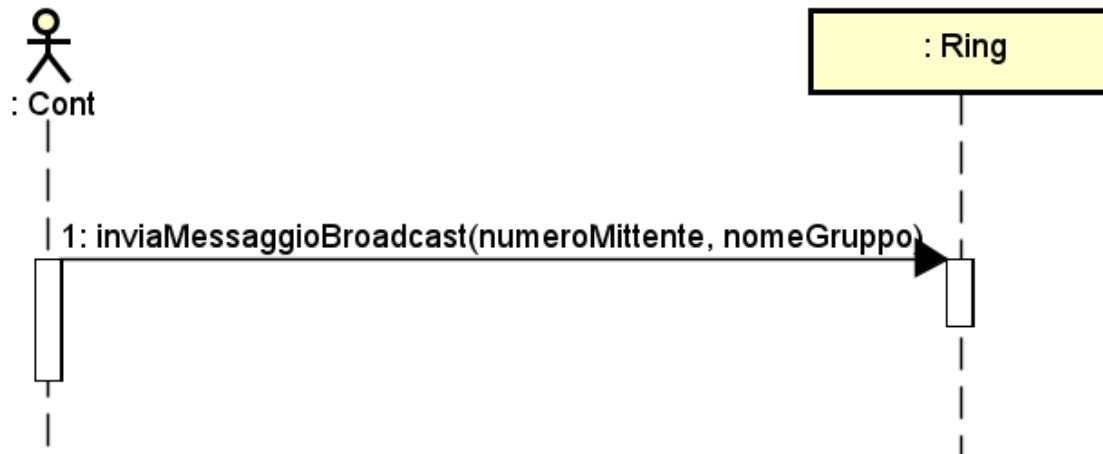
- UC8:



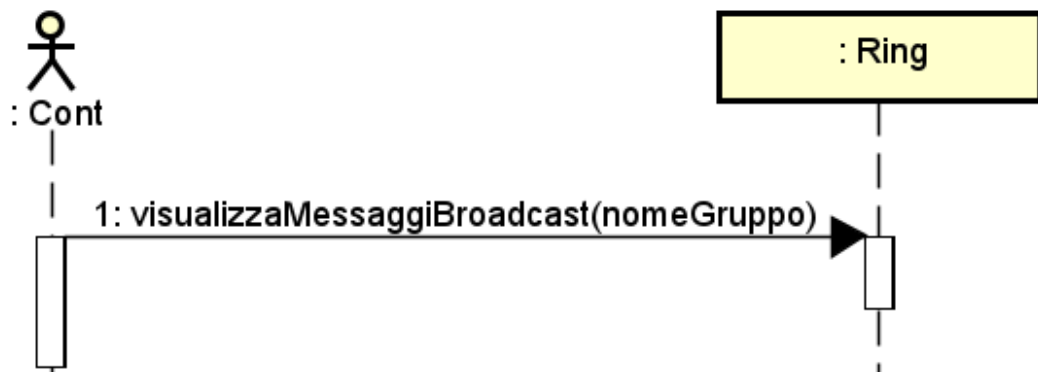
- UC9:



- UC10:



- UC11:



## 2.3 Contratti delle operazioni

### 2.3.1 Visualizza Messaggi

**Operazione:**

visualizzaMessaggi(numeroMittente)

**Riferimenti:**

Caso d'uso: Visualizza la cronologia messaggi

**Pre-condizioni:**

- Almeno un messaggio è stato registrato nel sistema

**Post-condizioni:**

### 2.3.2. Modifica Contatto

**Operazione:**

modificaContatto(numeroTelefono)

**Riferimenti:**

Caso d'uso: Gestione di un contatto

**Pre-condizioni:**

- E' in corso la modifica degli attributi della classe Contatto

**Post-condizioni:**

- È stata recuperata l'istanza Contatto da elencoContatti sulla base di numeroTelefono
- Gli attributi dell'istanza corrente Contatto sono stati modificati

### 2.3.3. Elimina Contatto

**Operazione:**

eliminaContatto(numeroTelefono)

**Riferimenti:**

Caso d'uso: Elimina Contatto

**Pre-condizioni:**

- È in corso l'eliminazione di un Contatto

**Post-condizioni:**

- È stata recuperata l'istanza Contatto da elencoContatti sulla base di numeroTelefono
- E' stato chiamato il metodo annullaPiano() per azzerare il credito
- L'istanza Contatto viene eliminata dalla lista elencoContatti

#### 2.3.4. Modifica Piano

**Operazione:**

modificaPianoContatto(numeroTelefono)

**Riferimenti:**

Caso d'uso: Gestione piano utente

**Pre-condizioni:**

- E' in corso la modifica del piano di un utente

**Post-condizioni:**

- È stata recuperata l'istanza Contatto da elencoContatti sulla base di numeroTelefono
- E' stato recuperato l'oggetto Ruolo Aziendale sulla base del testo inserito
- E' stato modificato il Ruolo dell'istanza Contatto con il nuovo
- E' stato aggiornato il credito del Contatto

#### 2.3.5 Visualizza Piano Residuo

**Operazione:**

visualizzaPianoResiduo(numeroMittente)

**Riferimenti:**

Caso d'uso: Visualizza il piano residuo

**Pre-condizioni:**

- Almeno un piano è stato registrato nel sistema

**Post-condizioni:**

#### 2.3.6 Crea Gruppo Utenti

**Operazione:**

creaGruppoUtenti()

**Riferimenti:**

Caso d'uso: Creazione di un gruppo di utenti

**Pre-condizioni:**

- E' in corso la creazione di un gruppo di utenti

**Post-condizioni:**

- E' stata creata la lista gruppoContatti
- È stata recuperata l'istanza Contatto da elencoContatti sulla base di numeroTelefono
- E' stato aggiunto il Contatto al gruppo creato
- E' stato aggiunto il gruppo creato a listaGruppi
- 

#### 2.3.7 Messaggio Gruppo

**Operazione:**

inviaMessaggioBroadcast(numeroMittente,nomeGruppo)

**Riferimenti:**

Caso d'uso: Messaggio di gruppo

**Pre-condizioni:**

- E' in corso l'invio di un messaggio broadcast

**Post-condizioni:**

- È stato recuperato il gruppo da listaGruppi sulla base del nome

- E' stato creato l'istanza messaggioCorrente Messaggio
- Gli attributi di messaggioCorrente sono stati inizializzati;
- Gli attributi di credito sono stati aggiornati;
- E' stato aggiunto il messaggioCorrente a elencoMessaggiBroadcast

### 2.3.8 Visualizza messaggi di gruppo

#### Operazione:

visualizzaMessaggiBroadcast(nomeGruppo)

#### Riferimenti:

Caso d'uso: Visualizza cronologia messaggi di gruppo

#### Pre-condizioni:

- Almeno un messaggio di gruppo è stato registrato nel sistema

#### Post-condizioni:

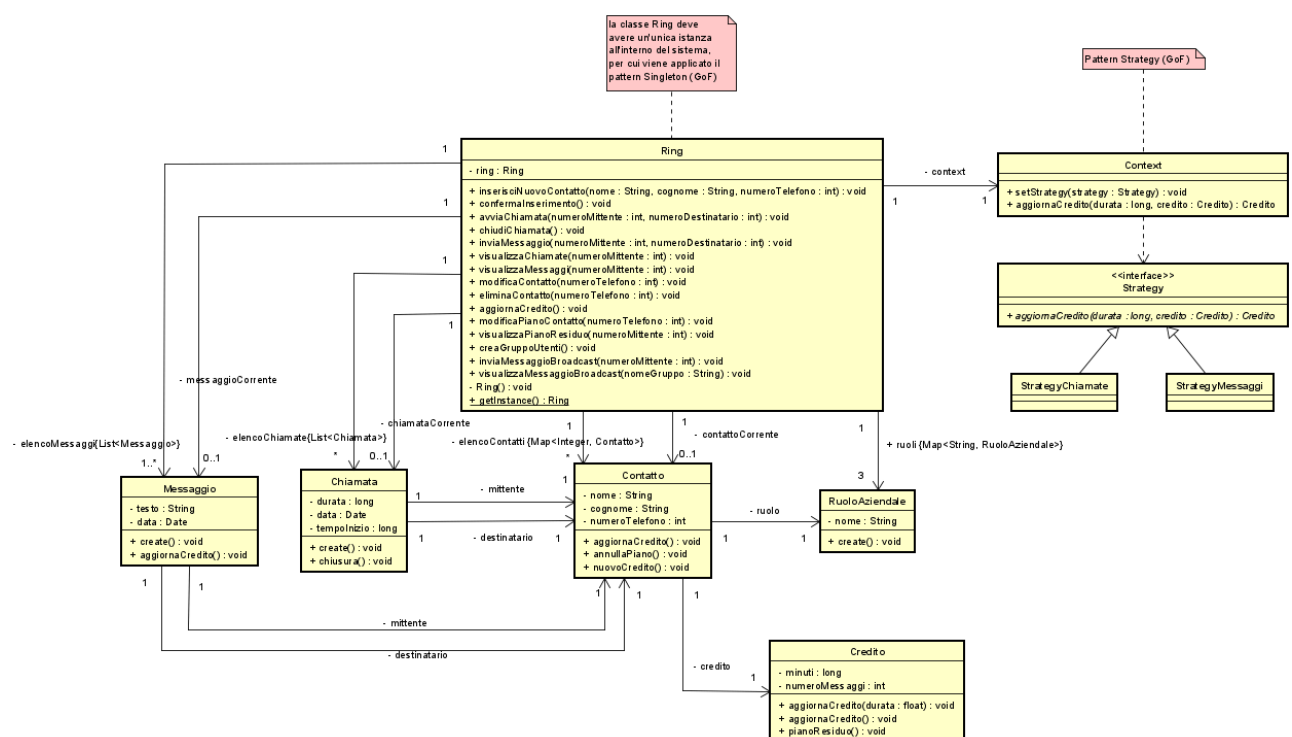
## 3. Progettazione

Nel progettazione implementiamo il pattern Strategy (GoF) per la gestione del credito delle chiamate e dei messaggi.

Il pattern strategy ci consente di aggiornare il credito in seguito ad una chiamata o ad un messaggio in maniera flessibile. Con l'ottica di ampliamento dell'applicazione attraverso il pattern Strategy sarebbe possibile creare anche una StrategyVideochiamate e una StrategyMessaggiMultimediali aumentando la flessibilità del codice.

### 3.1 Diagramma delle classi

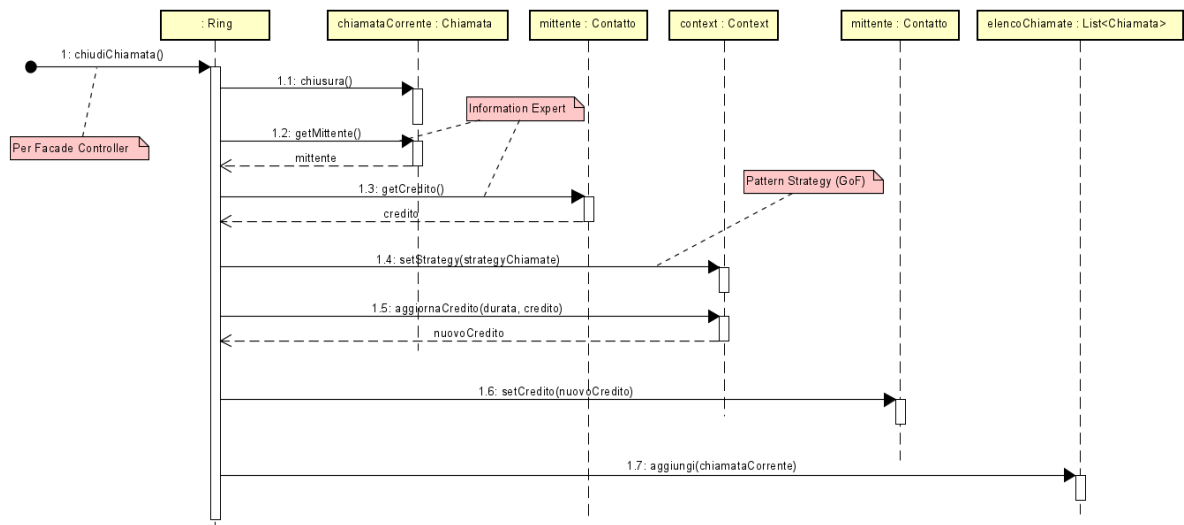
Il diagramma delle classi riflette l'implementazione del pattern Strategy. Con l'aggiunta di uno StrategyChiamate e uno StrategyMessaggi per la gestione del credito utente.



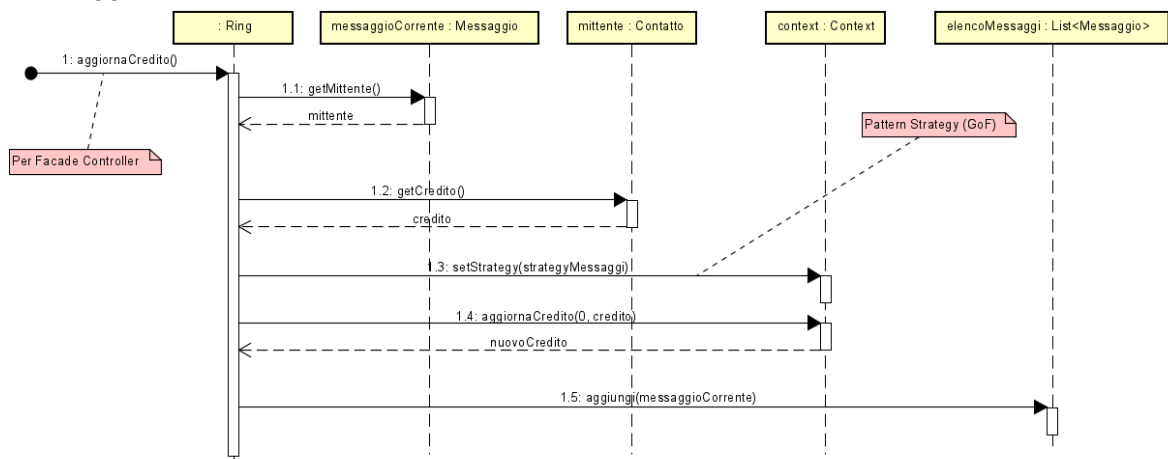
## 3.2 Diagrammi di sequenza

### 3.2.1 Diagrammi aggiornati tramite il pattern Strategy

- Chiudi Chiamata (SD4)

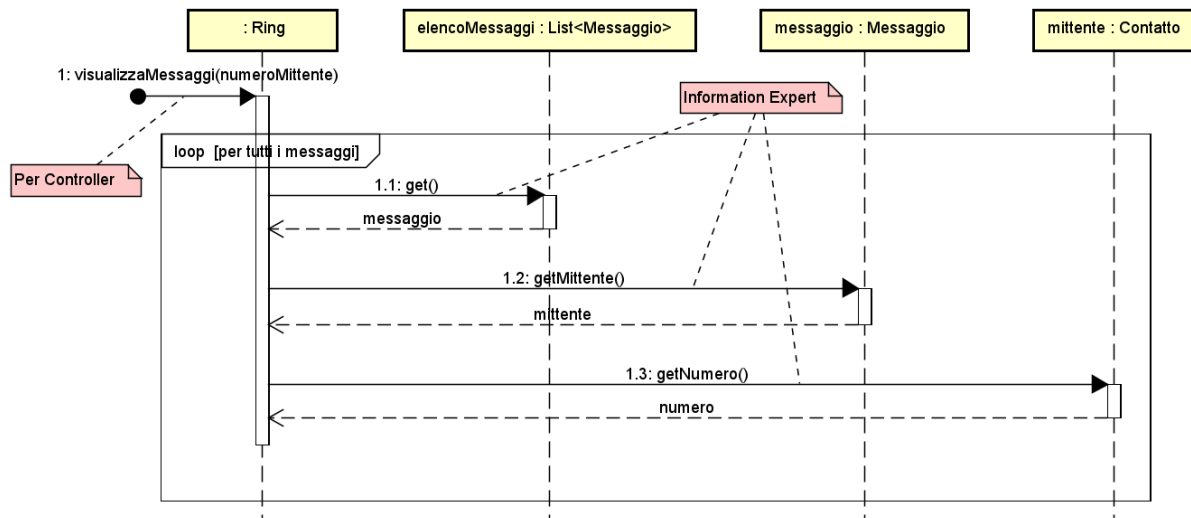


- Aggiorna Credito (SD6)

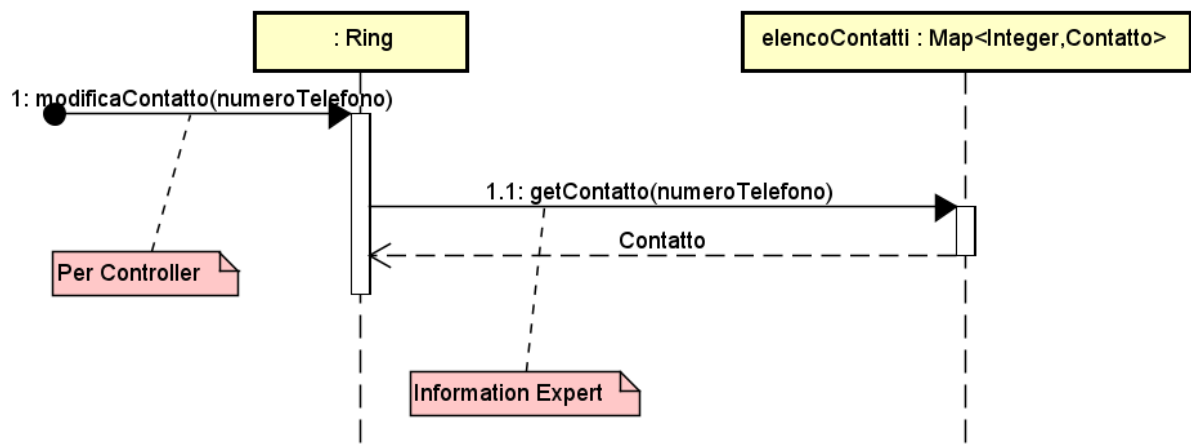


### 3.2.2 Diagrammi nuovi

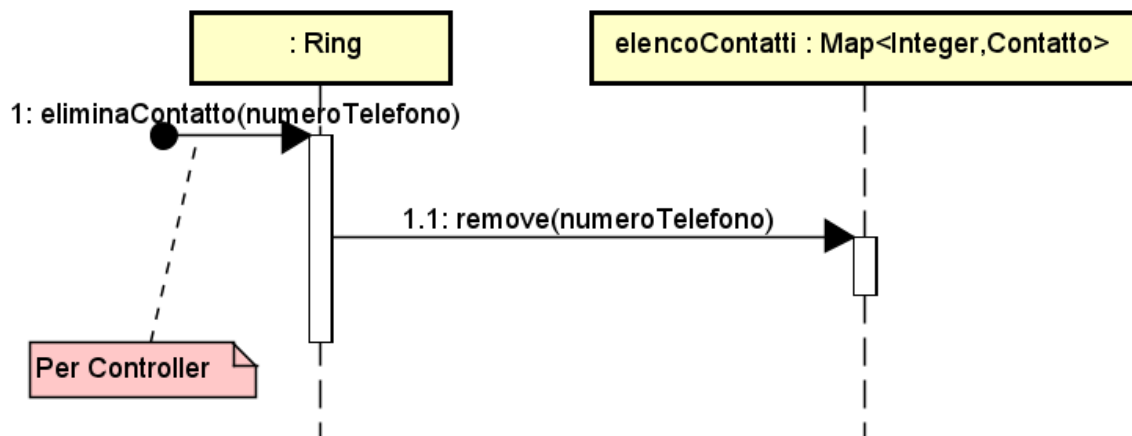
- Visualizza Messaggi (SD8)



- Modifica contatto (SD9)

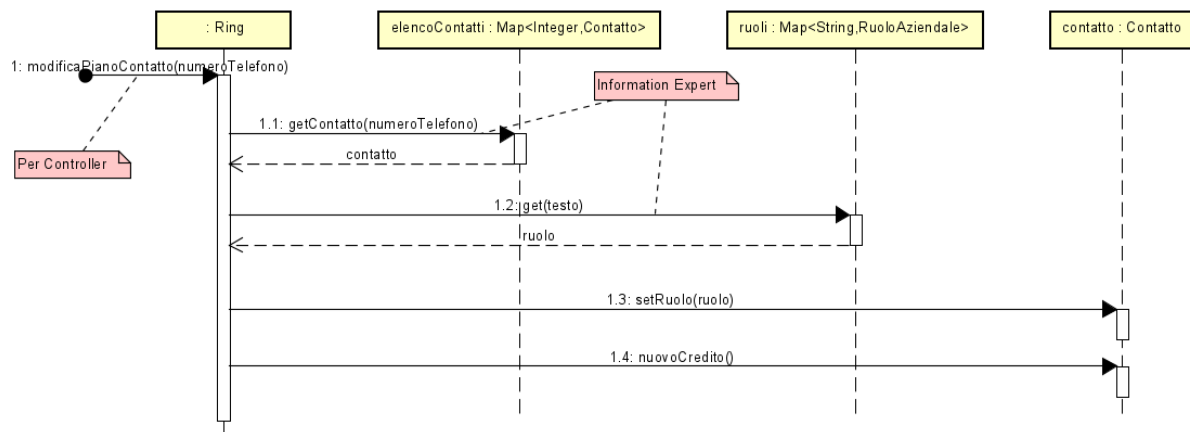


- Elimina contatto (SD10)

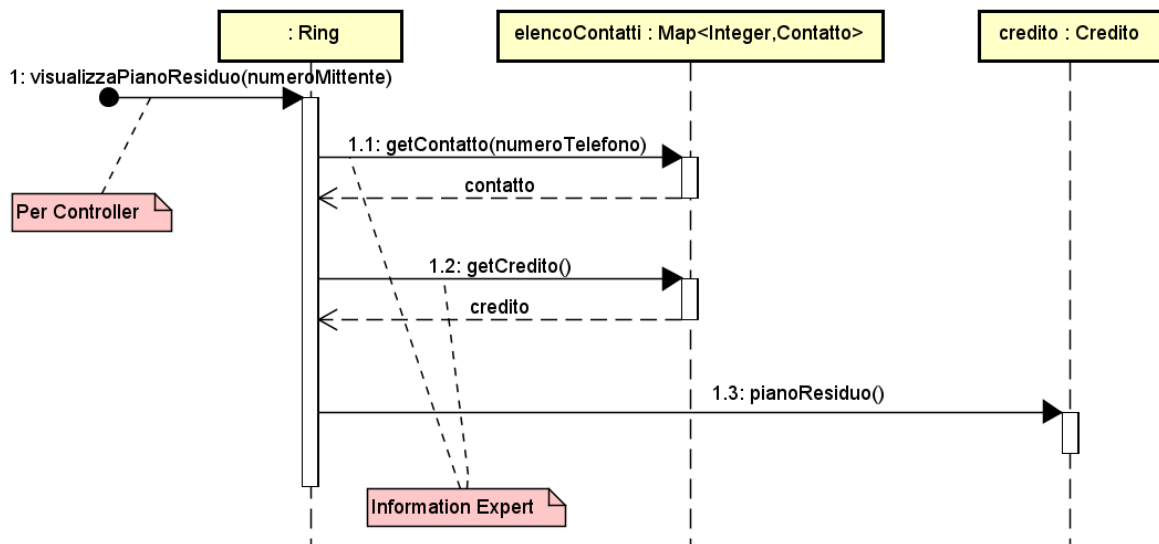




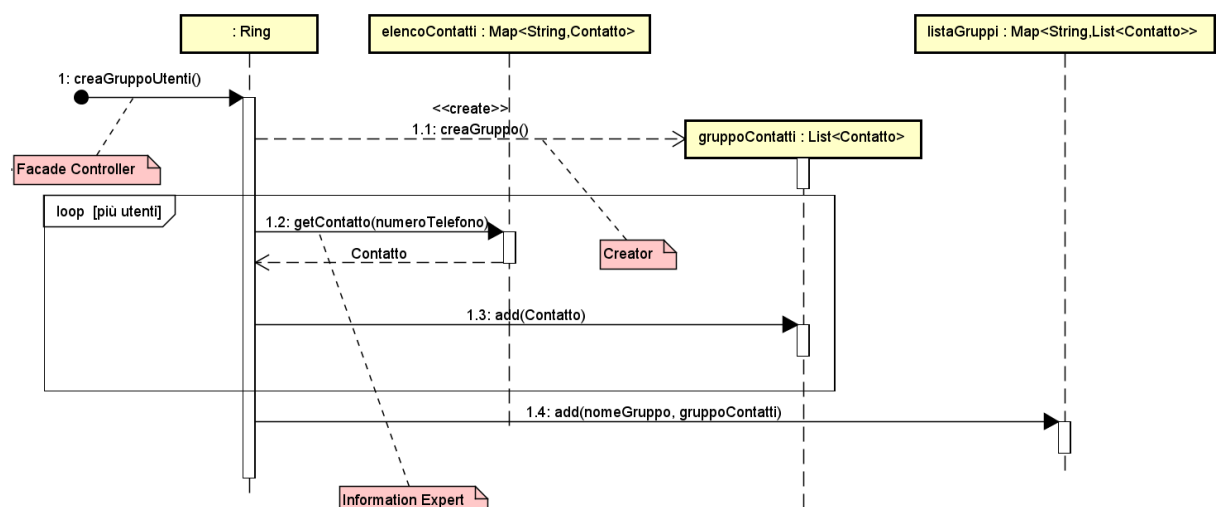
- Modifica piano contatto (SD11)



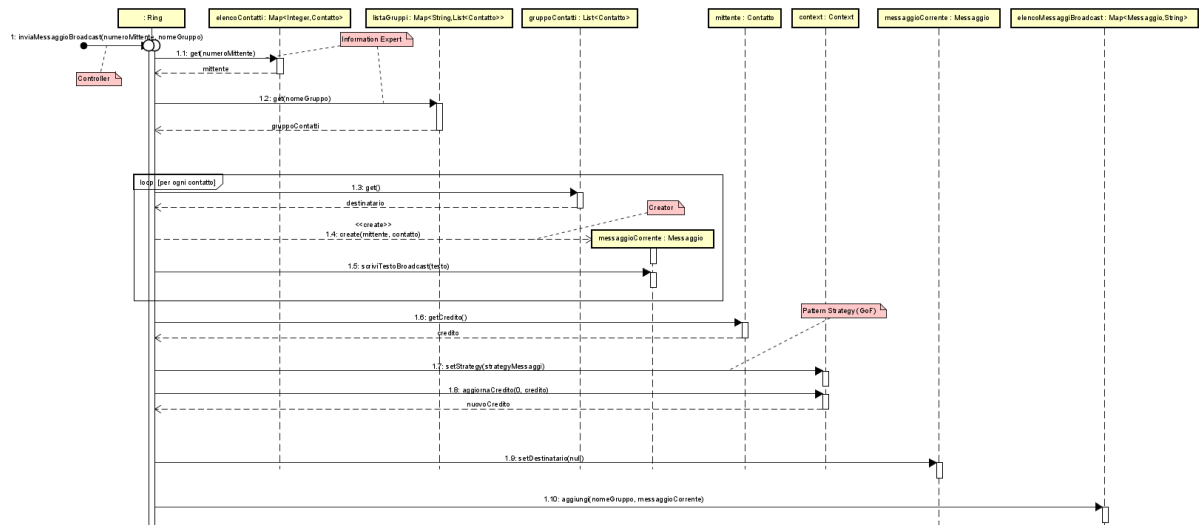
- Visualizza piano Residuo (SD12)



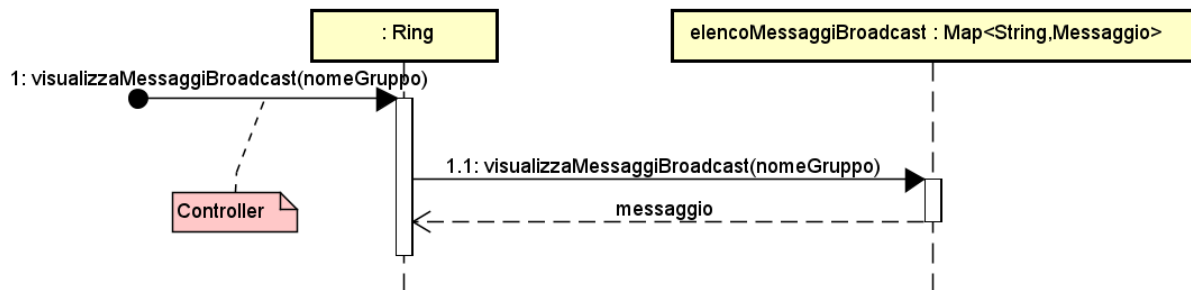
- Crea gruppo utenti (SD13)



- Invia messaggio broadcast (SD14)



- Visualizza l'ultimo messaggio broadcast (SD15)



### 3.3 Classi Test

#### 3.3.1 ChiamataTest

**Caso di Test:** testChiusura()

**Obiettivo:** Verificare che la durata della chiamata venga memorizzata correttamente.

**Fasi del Test:**

1. Creazione della chiamata.
2. Pausa del thread per 5 secondi.
3. Chiamata al metodo chiusura().
4. Verifica che la durata sia pari a 5 secondi.

#### 3.3.2 ContattoTest

**Caso di Test:** testNuovoCredito()

**Obiettivo:** Verificare che il credito inserito sia corretto.

**Fasi del Test:**

1. Creazione di un contatto con un proprio ruolo.
2. Cambiamento del ruolo tramite il metodo setRuolo().
3. Verifica dei minuti e del numeroMessaggi del nuovo ruolo.

#### 3.3.3 MessaggioTest

**Caso di Test:** testMessaggioBroadcast()

**Obiettivo:** Verifica che il testo del messaggio broadcast venga inserito correttamente

**Fasi del Test:**

1. Creazione di due nuovi utenti.
2. Chiamata al metodo scriviTestoBroadcast().
3. Verifica che il testo del messaggio inserito nel metodo sia corretto.

#### 3.3.4 RingTest

**Caso di Test:** testInserimentoContatto()

**Obiettivo:** Verificare che un nuovo contatto venga inserito correttamente nel campo contattoCorrente

**Fasi del Test:**

1. Creazione dell'istanza ring.
2. Chiamata del metodo inserisciNuovoContatto().
3. Verifica che il contatto sia stato inserito nel campo contattoCorrente

**Caso di Test:** testConfermaInserimento()

**Obiettivo:** Verificare che il nuovo contatto venga inserito correttamente nella lista contatti

**Fasi del Test:**

1. Creazione dell'istanza ring.
2. Chiamata del metodo confermaInserimento().
3. Verifica che il contatto sia stato inserito nell'elenco contatti.

#### 3.3.5 StrategyTest

**Caso di Test:** testAggiornaCreditoMessaggi()

**Obiettivo:** Verificare che il credito dei messaggi venga aggiornato

**Fasi del Test:**

1. Creazione del context
2. Registrazione della data di oggi
3. Chiamata del metodo aggiornaCredito()
4. Verifica che il credito dei messaggi venga aggiornato

**Caso di Test:** testAggiornaCreditoChiamate()

**Obiettivo:** Verificare che il credito delle chiamate venga aggiornato

**Fasi del Test:**

5. Creazione del context
6. Registrazione della data di oggi
7. Chiamata del metodo aggiornaCredito()
8. Verifica che il credito delle chiamate venga aggiornato