

PawBookings: Elaborazione - Iterazione 5

Giuseppe Leocata, Alberto Provenzano, Daniele Lucifora

Introduzione

Per l'iterazione 5, sono stati scelti i seguenti requisiti:

- Implementazione delle regole di dominio.
- Implementazione di ulteriori estensioni relative all'UC2.
- Aggiunta estensione al caso d'uso UC8 (Progettata ma non implementata a livello di codice).
- Implementazione del caso d'uso d'avviamento necessario per inizializzare questa iterazione.
- Dati solo in memoria principale.

Aggiornamenti elaborati della fase di Ideazione

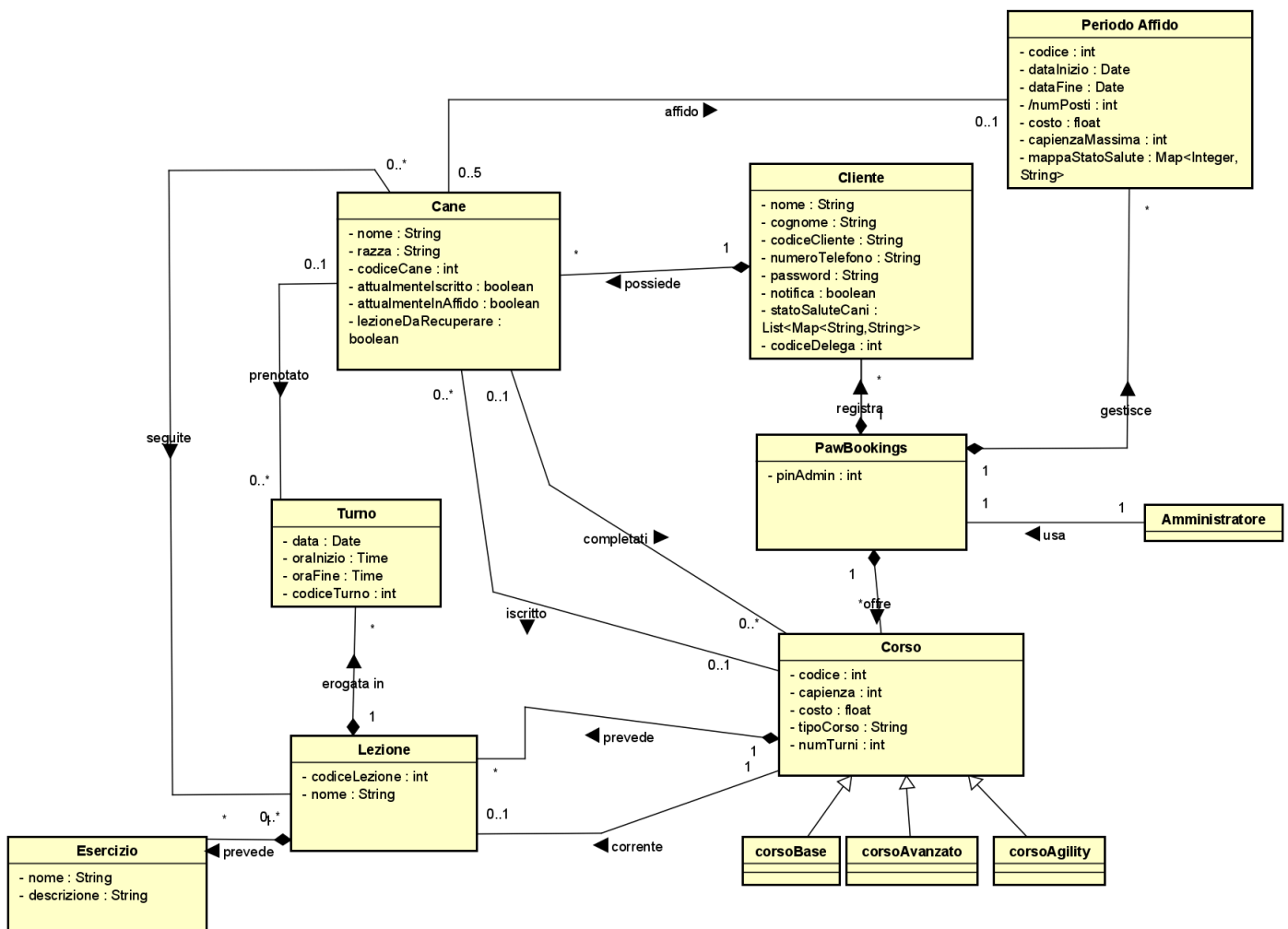
Sono stati aggiornati i documenti riguardanti le Regole di Dominio rendendo più chiari alcuni passaggi ed il Modello dei casi d'uso. In quest'ultimo, nello specifico, è stato aggiunto lo scenario alternativo di rimozione di un turno (UC8), rinominando l'intero caso d'uso "Gestisci turni".

1. Analisi Orientata agli Oggetti

Proseguendo con lo stesso approccio utilizzato nelle iterazioni precedenti, verranno riproposti: Modello di Dominio, SSD e Contratti delle Operazioni.

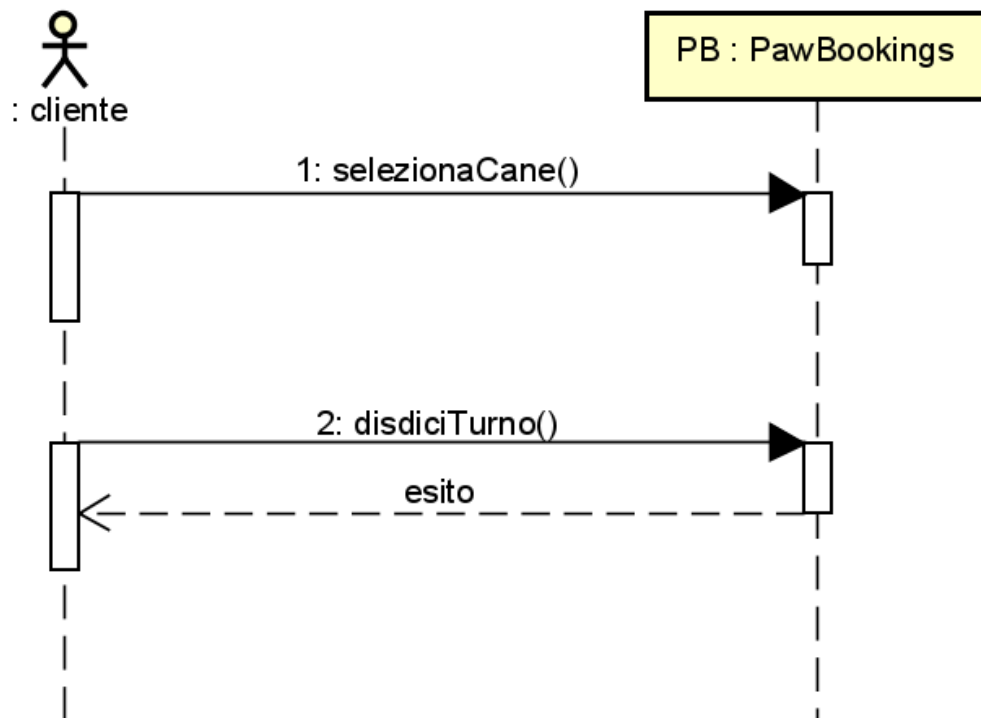
1.1. Modello di Dominio

Rimasto pressoché invariato. L'unica differenza sta nell'aggiunta dell'attributo *lezioneDaRecuperare* in **Cane**.

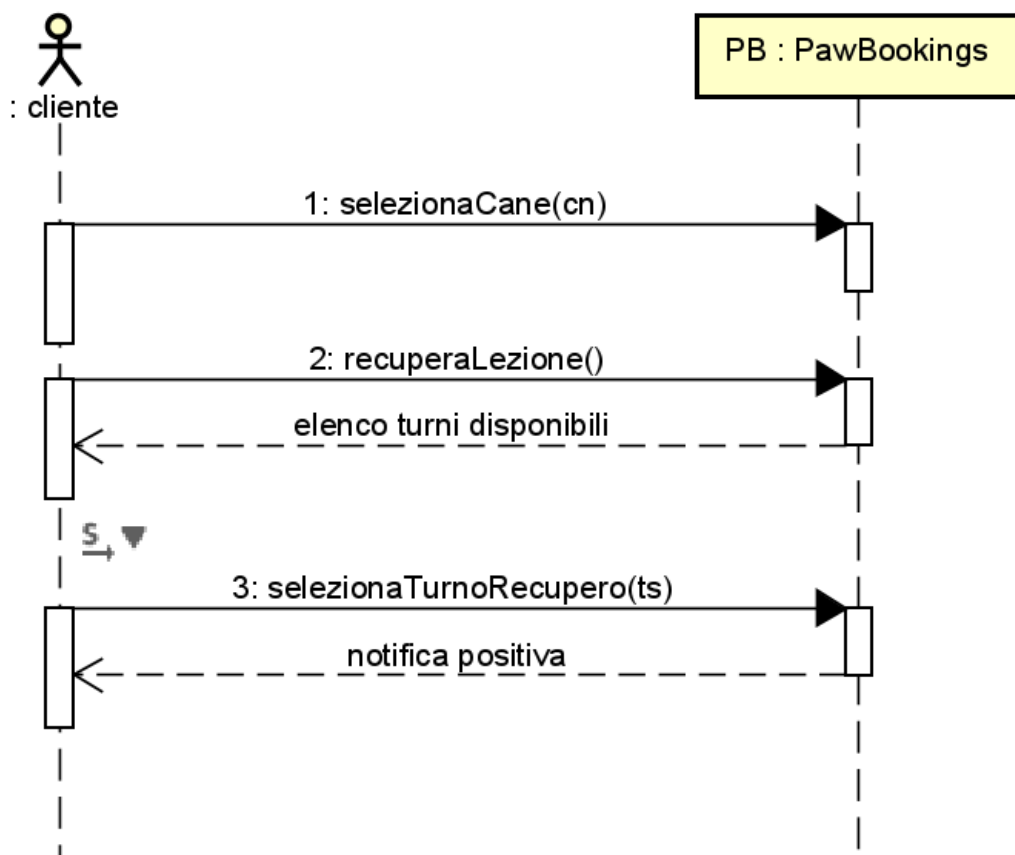


1.2. Diagrammi di Sequenza di Sistema (SSD)

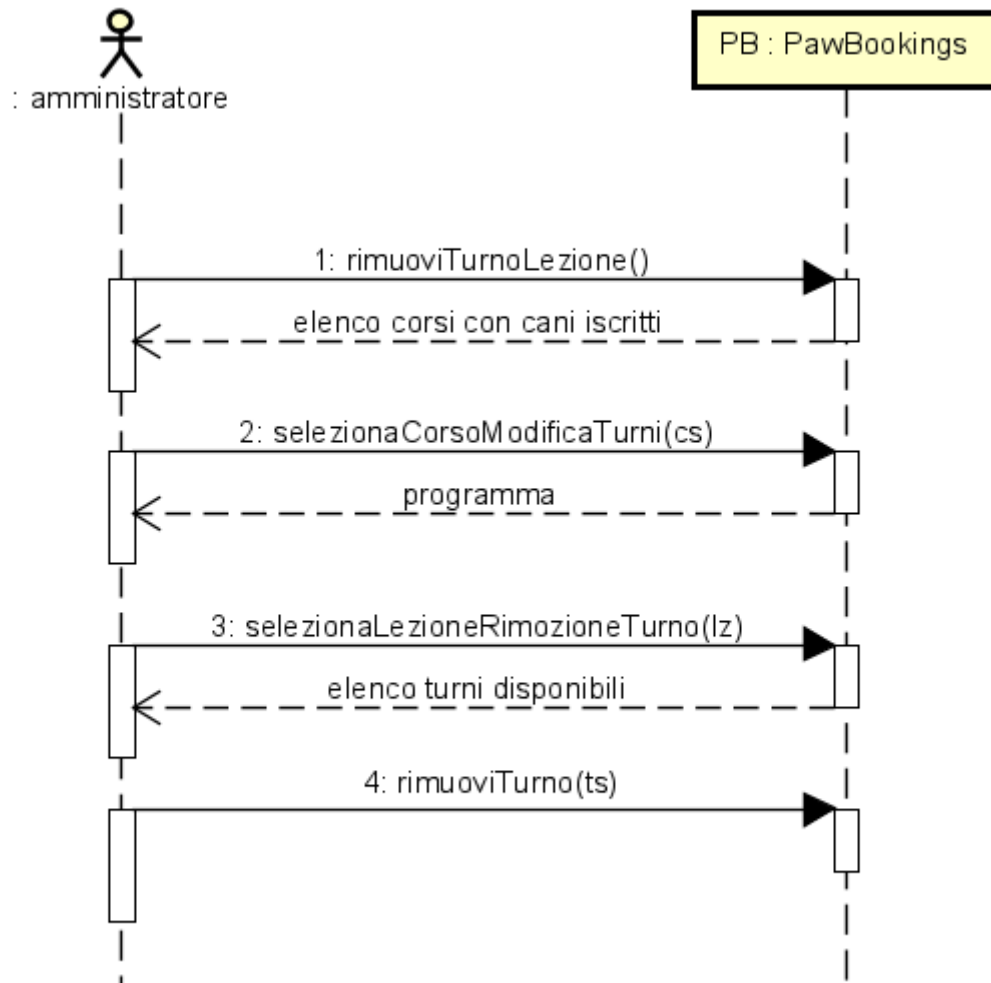
1.2.1. SSD UC2 - Estensione 1



1.2.2. SSD UC2 - Estensione 2



1.2.3. SSD UC8 - Estensione 1



1.3. Contratti delle Operazioni

1.3.1. C01: disdiciTurno

operazione:	disdiciTurno()
referimenti:	Caso d'uso UC2: Gestisci prenotazione turno lezione
pre-condizioni:	- è stata recuperata l'istanza cn di Cane - è stata recuperato il Turno t associato a cn tramite l'associazione "prenotato"
post-condizioni:	- cn è dissociato da t

1.3.2. C01: selezionaTurnoRecupero

operazione:	selezionaTurnoRecupero(ts: Turno)
referimenti:	Caso d'uso UC2: Gestisci prenotazione turno lezione
pre-condizioni:	- è stata recuperata l'istanza cn della classe Cane selezionata dall'utente
post-condizioni:	- è stata aggiornata l'associazione "prenotato" tra ts e cn

2. Progettazione Orientata agli Oggetti

2.1. Pattern applicati

E' stato applicato il pattern GoF Composite per il calcolo dei vari importi dovuti dal Cliente sia per quanto riguarda i corsi d'addestramento (pagamento del corso ed eventualmente di lezioni da recuperare) che per gli affidi. Si è scelto questo pattern in quanto si possono presentare situazioni nelle quali le diverse politiche di sconto/prezzo si sovrappongono.

Sono stati, inoltre, applicati i pattern GRASP (Creator, Information Expert, Controller, Low Coupling e High Cohesion e Pure Fabrication). Il pattern GRASP pure fabrication è stato utilizzato per la creazione delle classi relative alle politiche di sconto/prezzo per avere un'alta coesione e un accoppiamento basso.

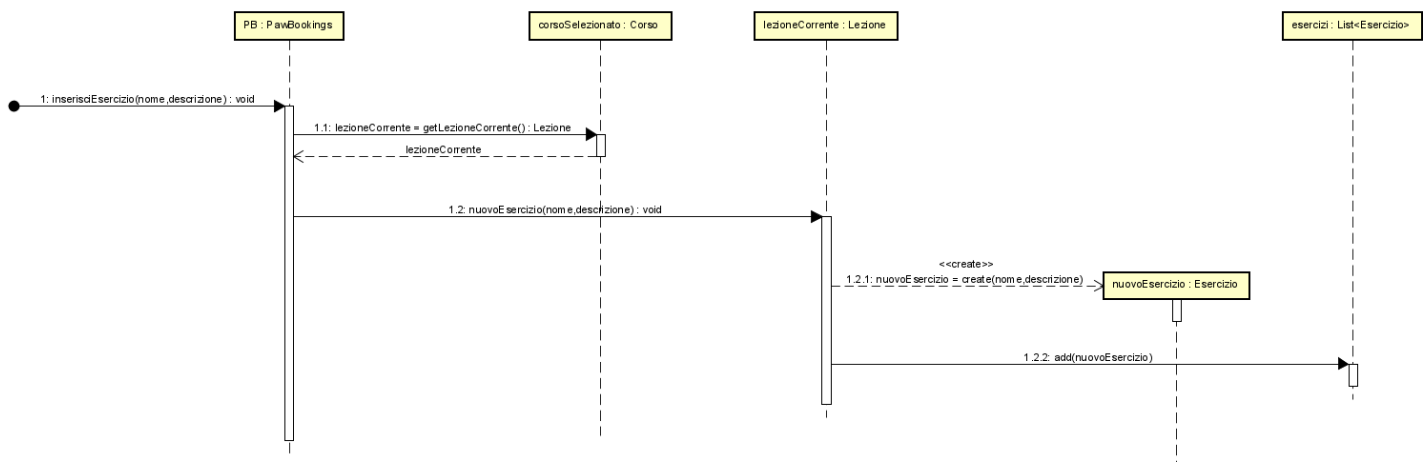
2.2. Diagrammi di Sequenza (SD)

2.2.1. SD delle iterazioni precedenti aggiornati

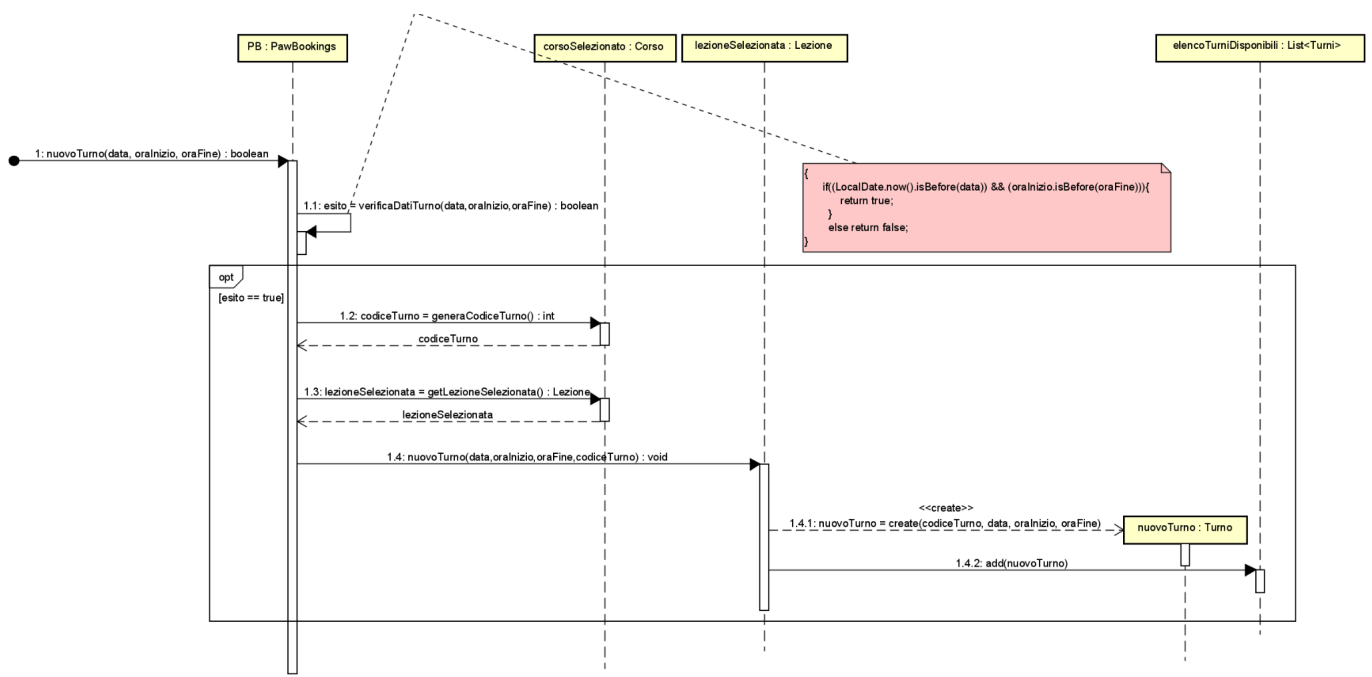
2.2.1.1. UC4 SD2: confermaConclusioneAffido

Per la visione di questo diagramma, si invita il lettore ad aprire il file *"iterazione5.asta"*.

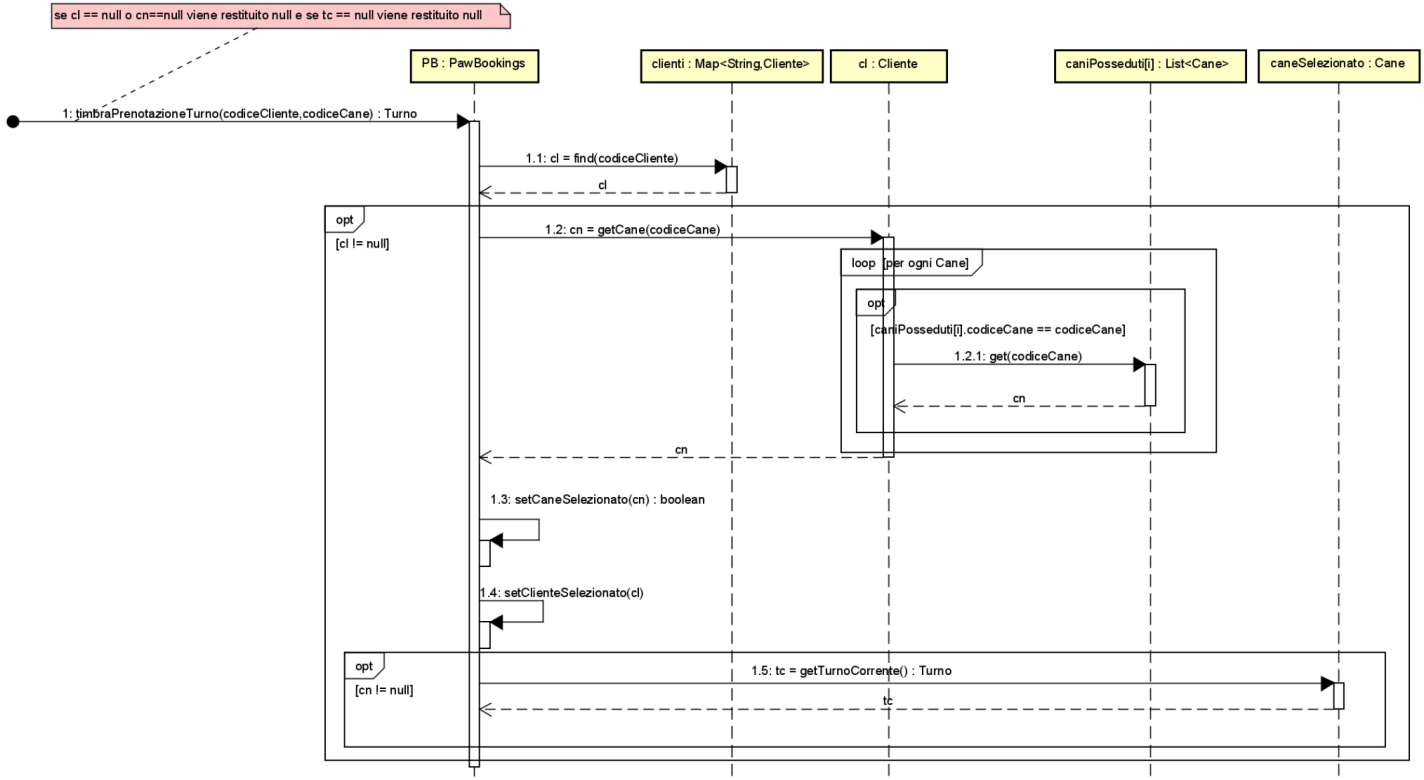
2.2.1.2. UC7 SD8: inserisciEsercizio



2.2.1.3. UC8 SD4: nuovoTurno



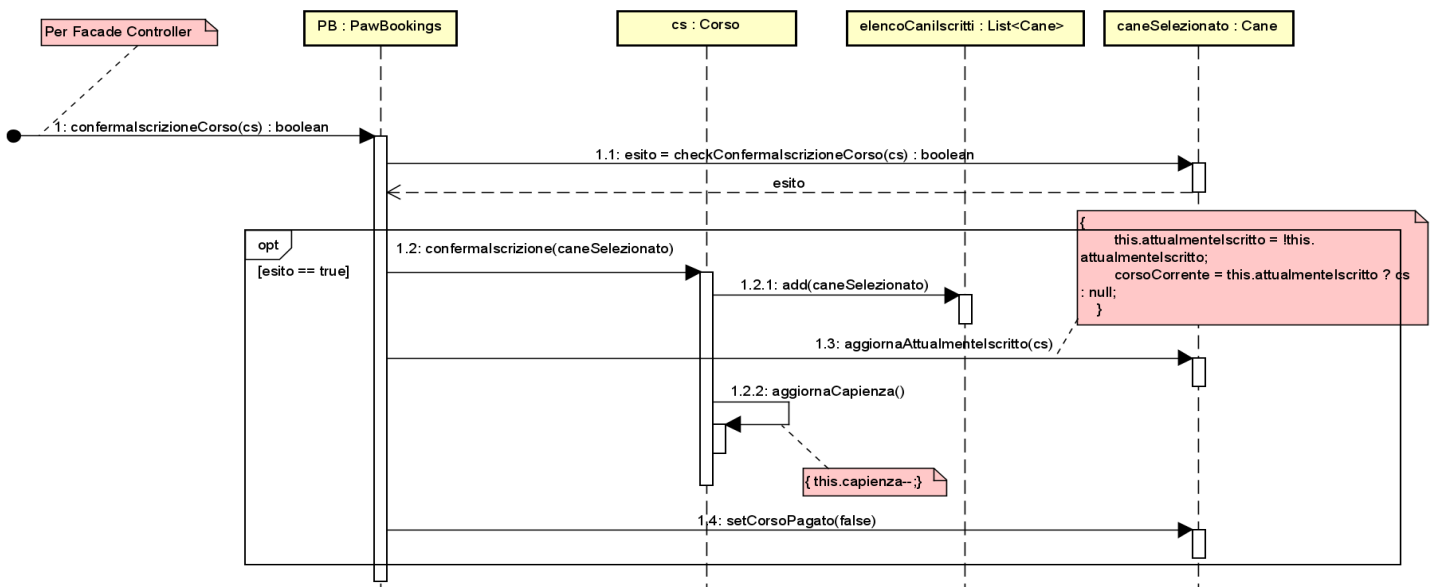
2.2.1.4. UC14 SD1: timbraPrenotazioneTurno



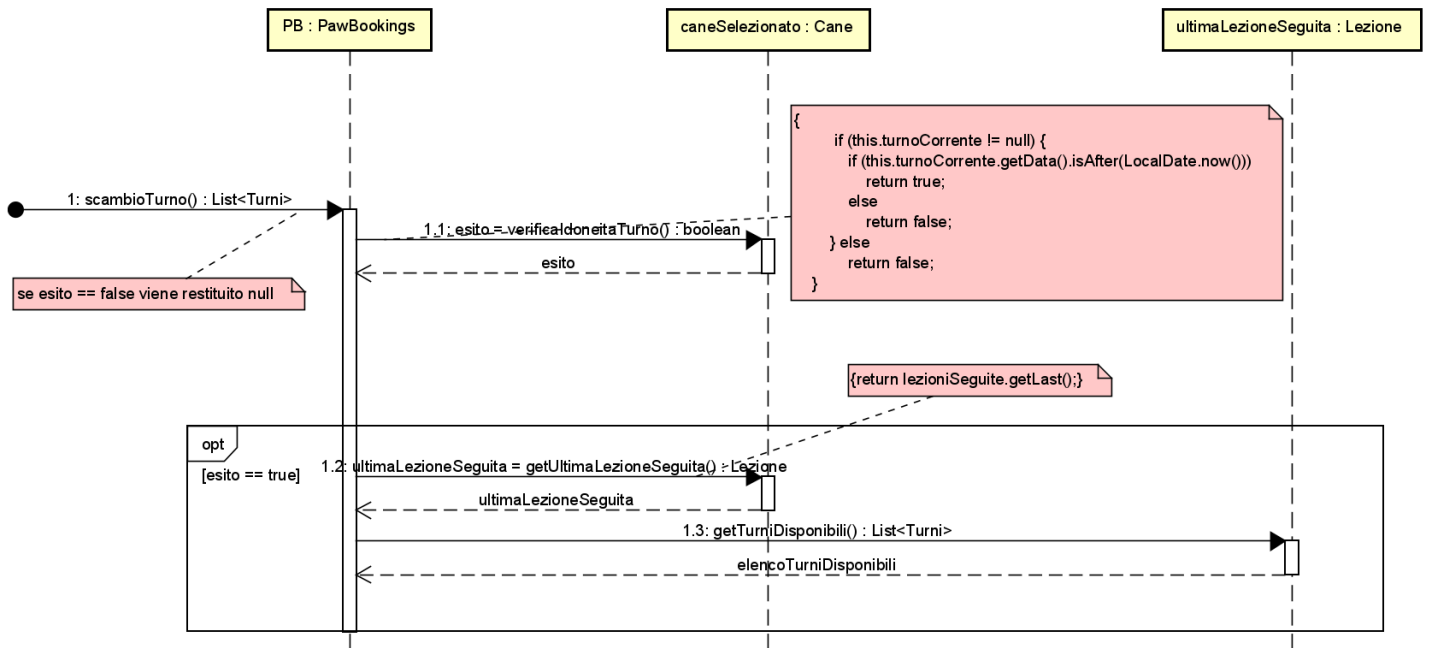
2.2.1.5. UC14 SD2: confermaTimbroTurno

Per la visione di questo diagramma, si invita il lettore ad aprire il file “*iterazione5.asta*”.

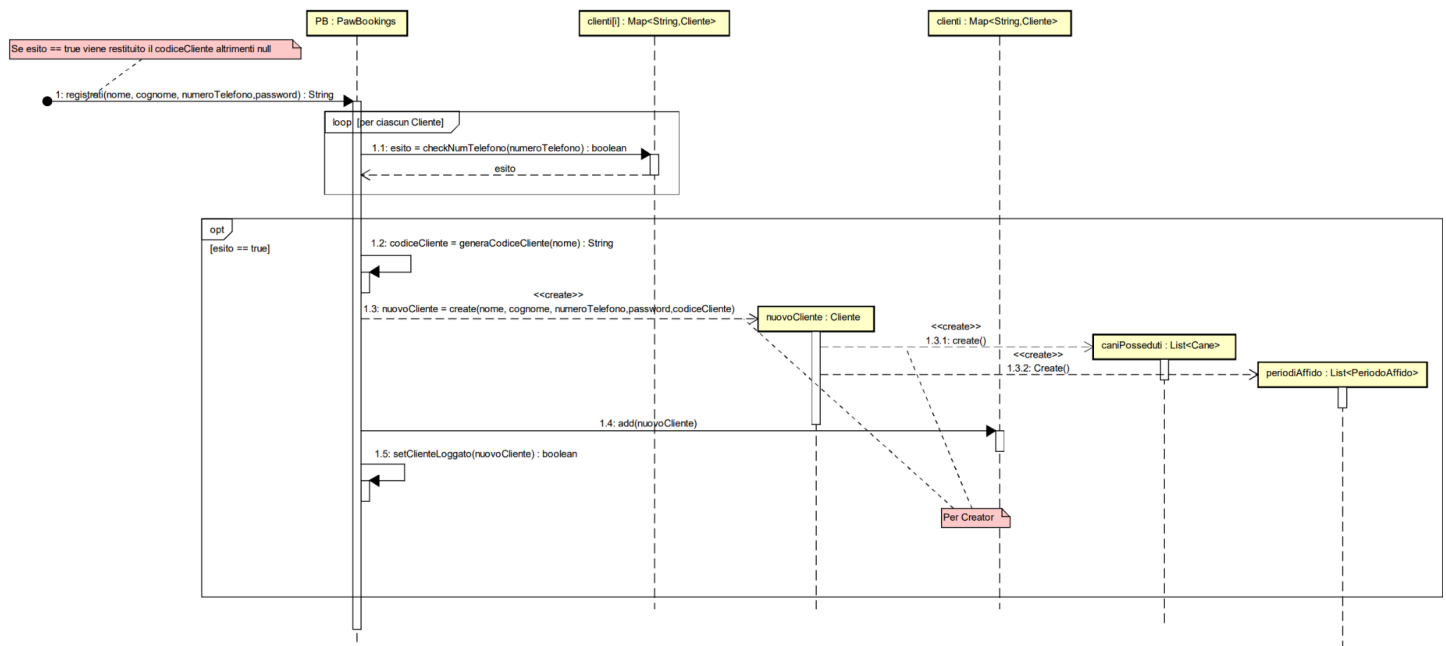
2.2.1.6. UC1 SD2 confermalscrizioneCorso



2.2.1.7. UC9 SD1: scambioTurno

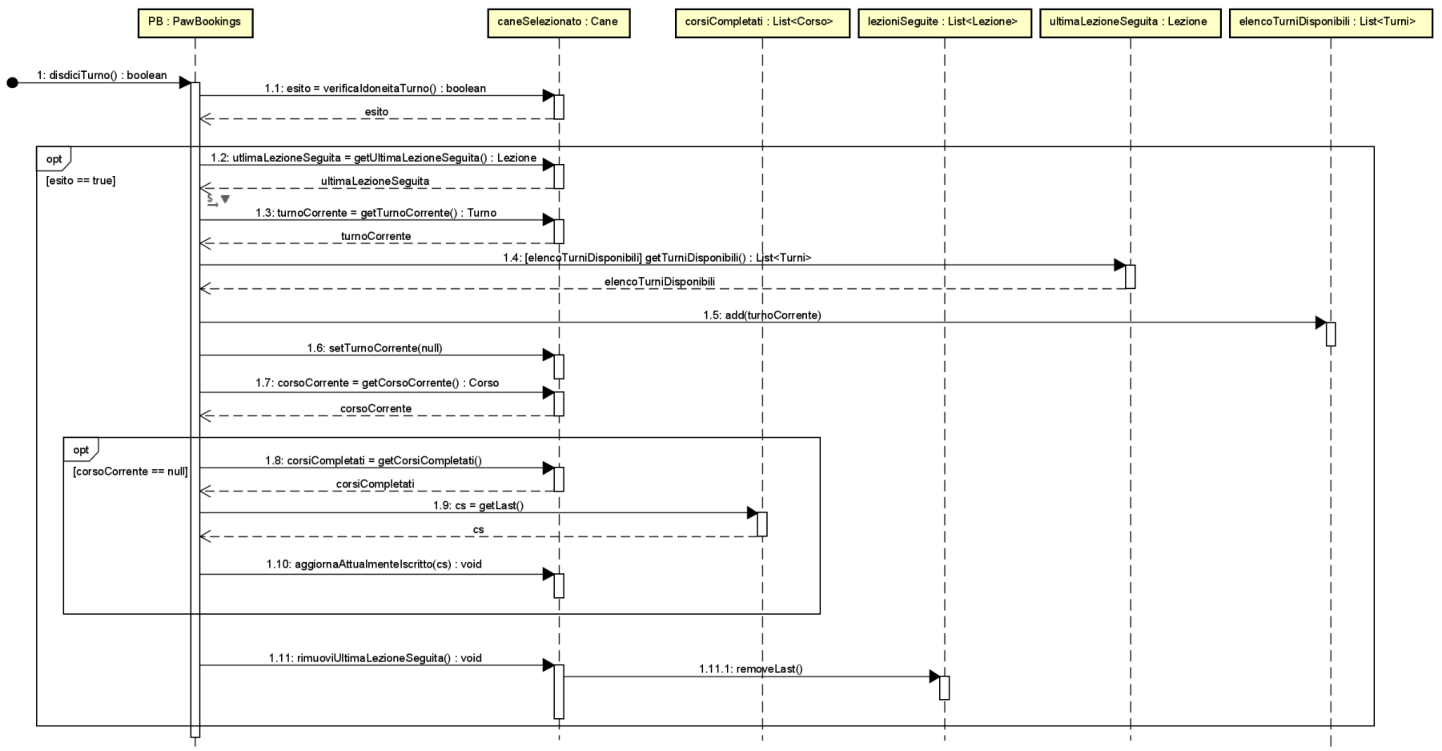


2.2.1.8. UC5 SD2: registrati

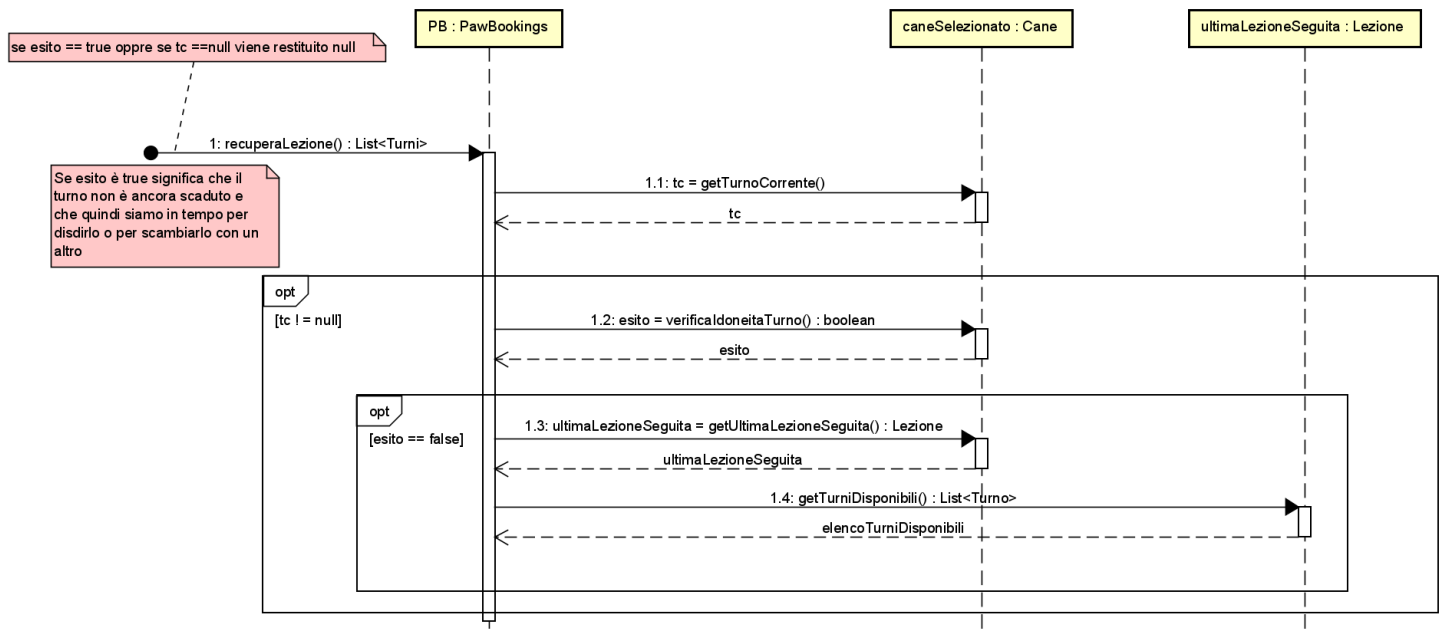


2.2.2. Nuovi SD della quinta iterazione

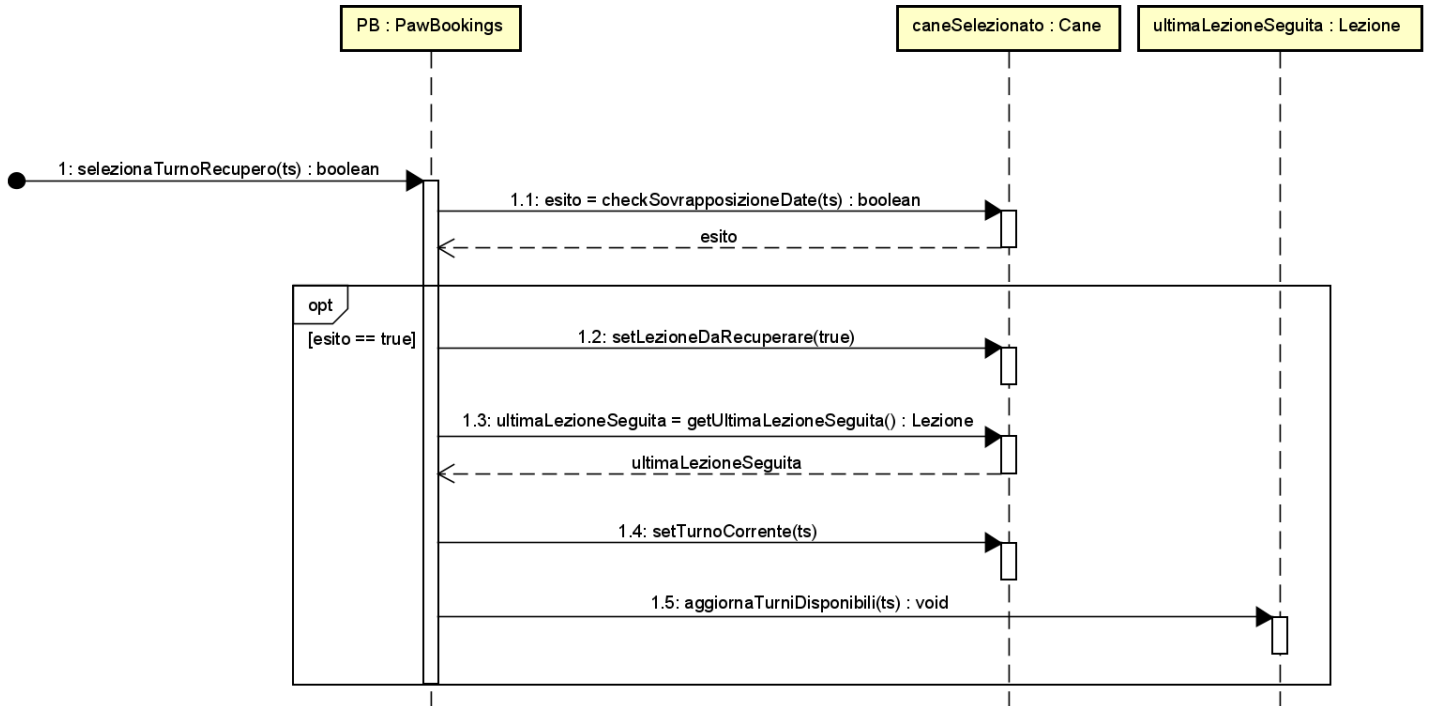
2.2.2.1. UC2 SD3: disdiciTurno



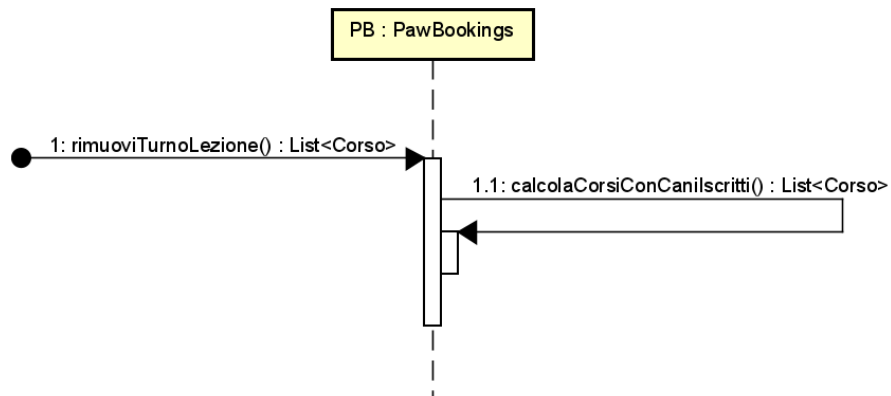
2.2.2.2. UC2 SD4: recuperaLezione



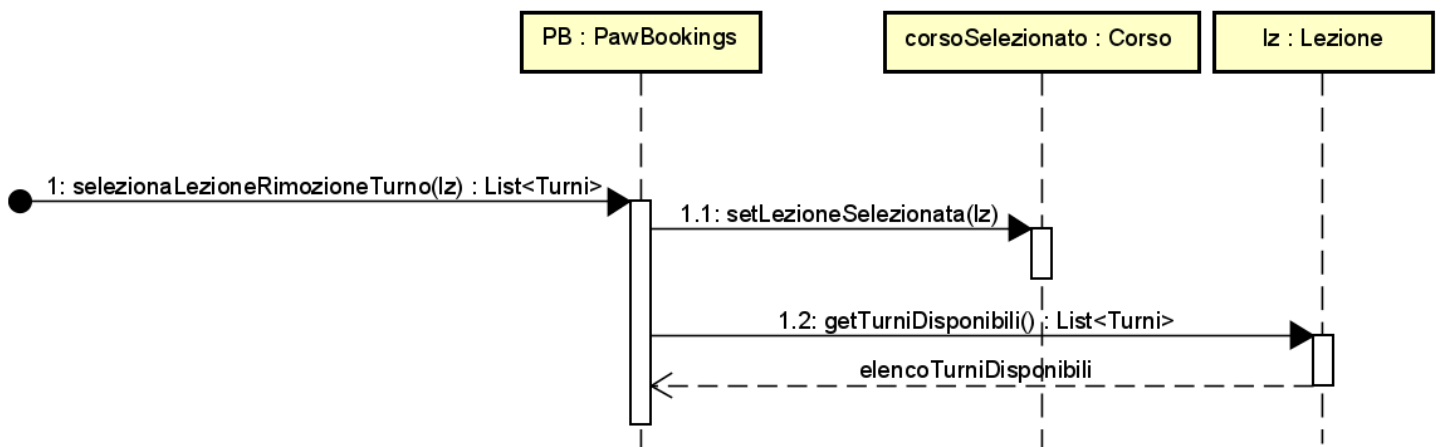
2.2.2.3. UC2 SD5: selezionaTurnoRecupero



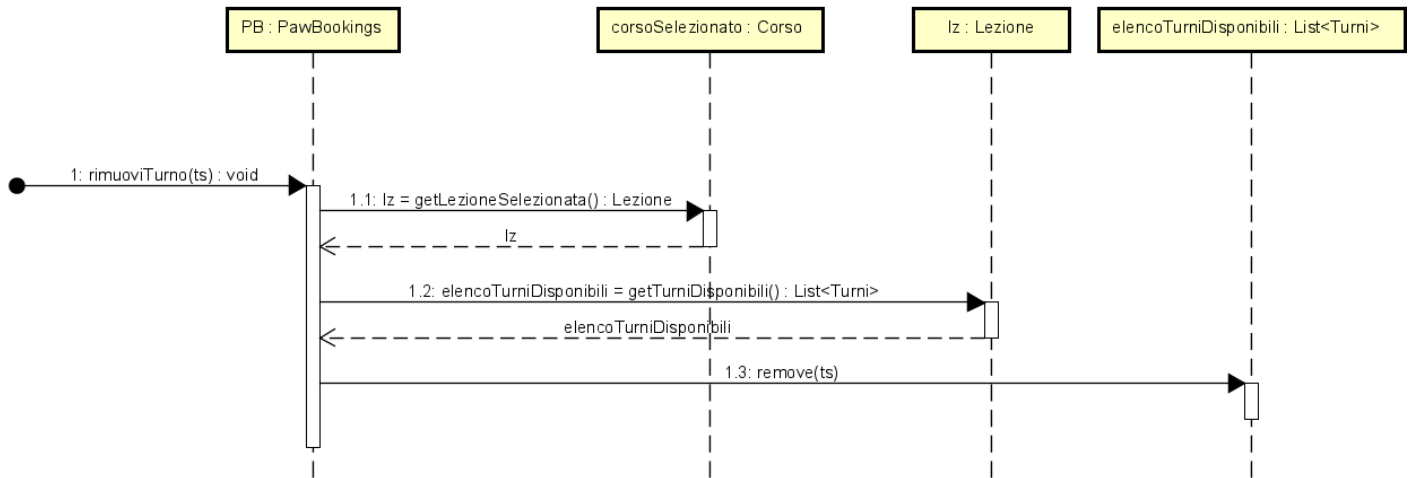
2.2.2.4. UC8 SD5: rimuoviTurnoLezione



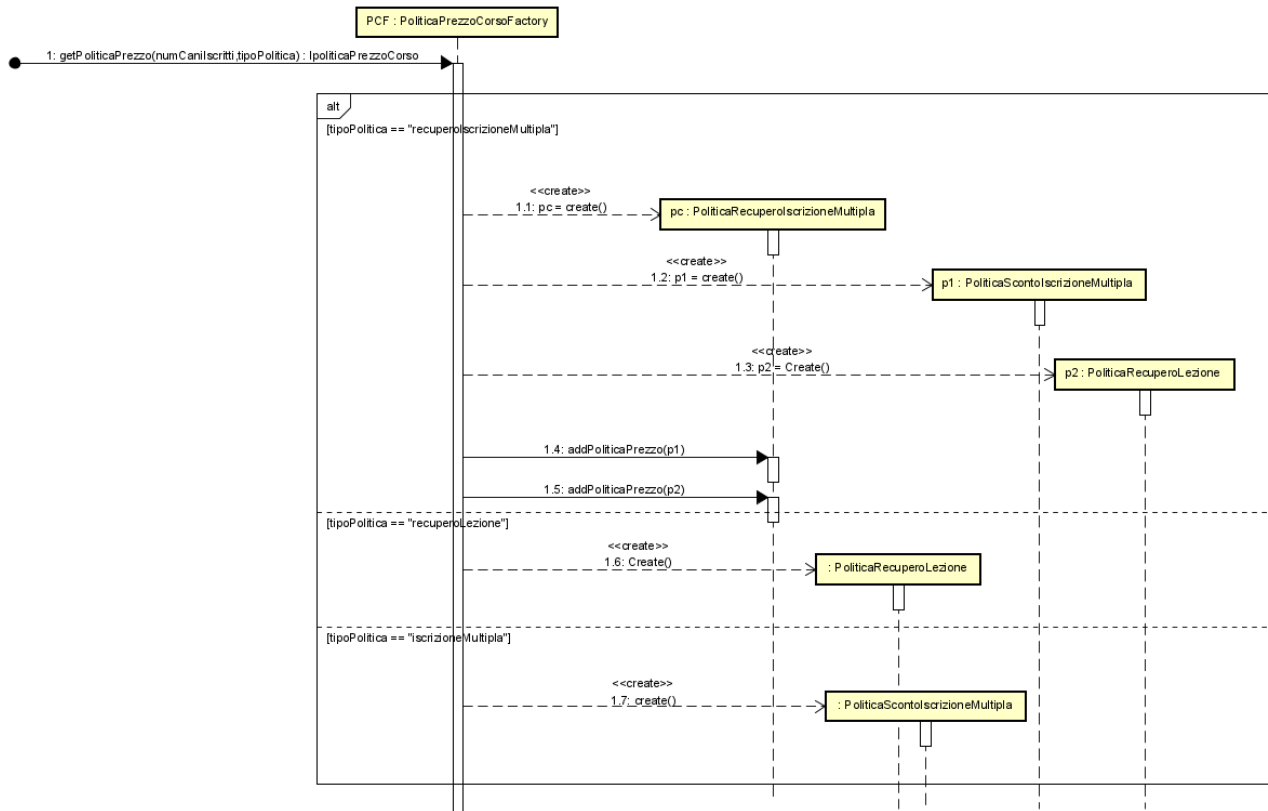
2.2.2.5. UC8 SD6: selezionaLezioneRimozioneTurno



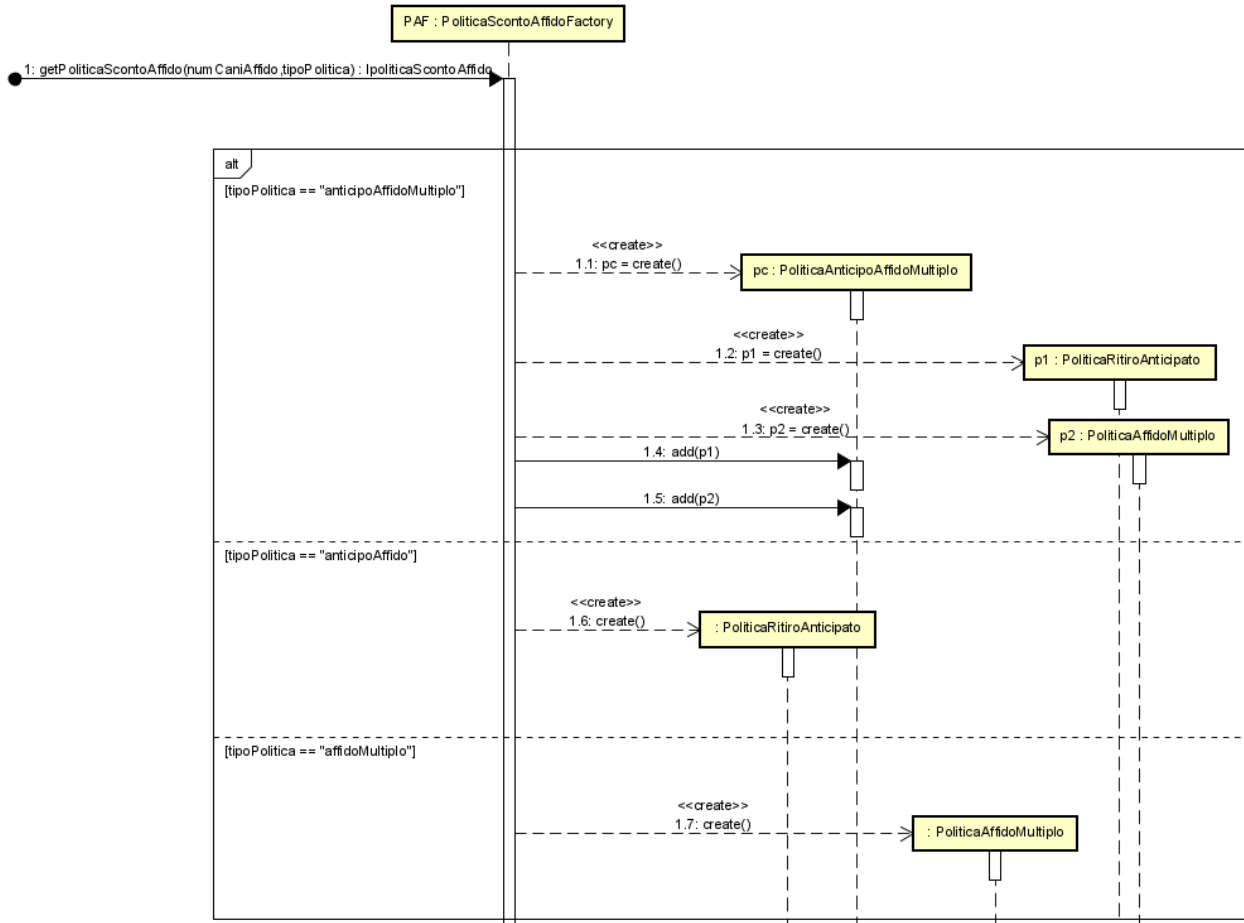
2.2.2.6. UC8 SD7: rimuoviTurno



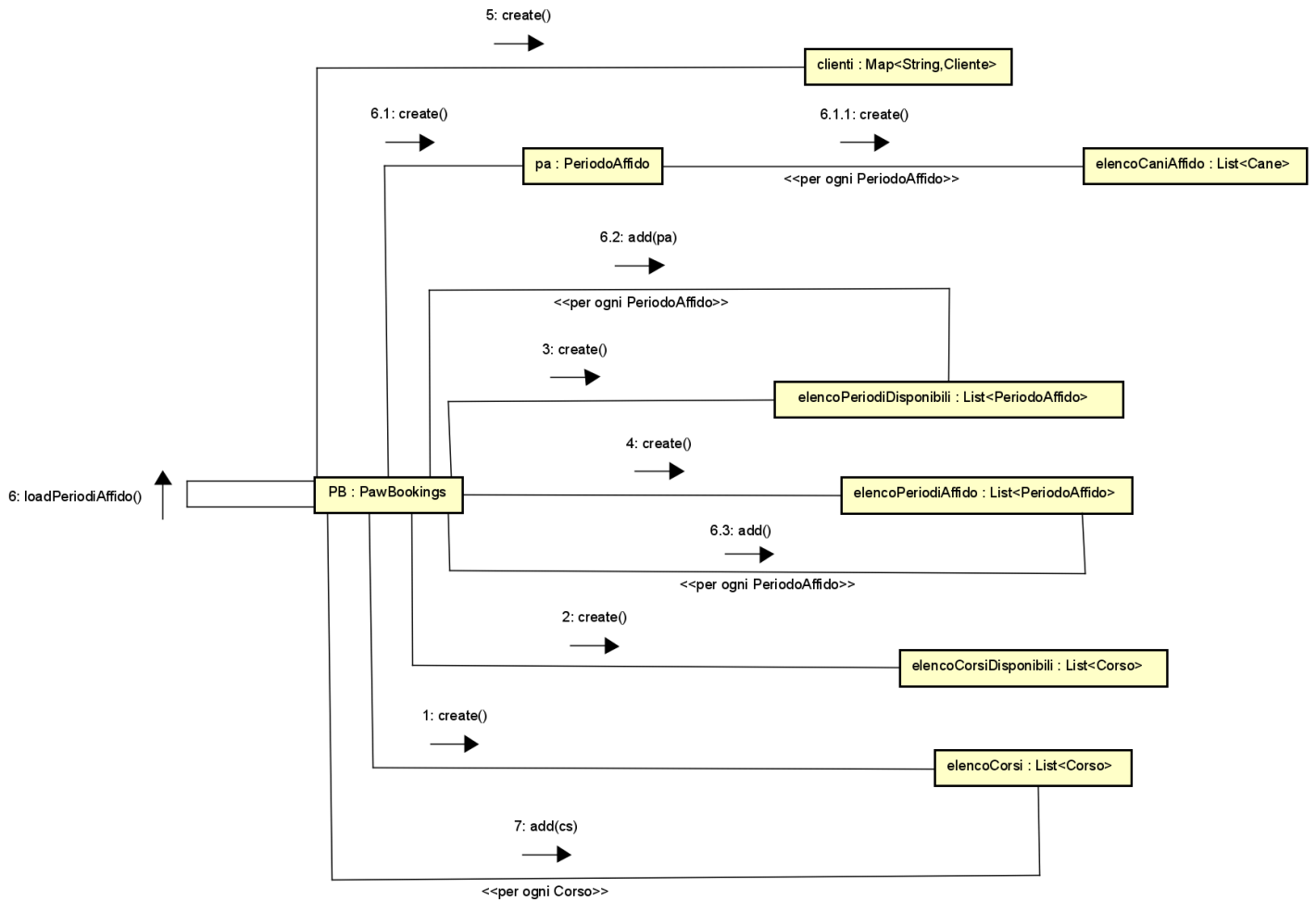
2.2.2.7. getPoliticaPrezzo



2.2.2.8. getPoliticaSconto



2.2.2.9. Caso d'uso d'avviamento



2.3. Diagramma delle Classi di Progetto

Per la visione del DCD, si invita il lettore ad aprire il file *“iterazione5.asta”*.

3. Testing

Nella quinta iterazione è stato scelto di concentrare il testing esclusivamente nella classe PawBookings implementando dei test mirati ad evidenziare il corretto funzionamento del software. È stato utilizzato un criterio di scelta dei metodi da testare che, per questa quinta iterazione, tenga conto del flusso degli scenari alternativi 3c e 3d del caso d'uso UC2 e del corretto funzionamento del software relativamente alle regole di dominio.

Per quinta iterazione sono state aggiunte solamente due nuove classi di test:

- PawBookingsTest6: classe di test che comprende i test unitari relativi agli scenari 3c e 3d del caso d'uso UC2
- PawBookingsTest7: classe di test che comprende i test unitari relativi alle regole di dominio.

Inoltre, relativamente alla classe Cane è stato introdotto il test relativamente al metodo `verificaDoneitaTurno`.

I test unitari implementati per le precedenti iterazioni sono stati adattati alle modifiche introdotte dalla nuova iterazione e sono funzionanti.

In particolar modo sono stati individuati i seguenti metodi delle relative classi:

- PawBookings
 - `disdiciTurno`
 - viene verificato che una volta chiamato il metodo sotto test il turno corrente del cane selezionato diventi **null**.
 - viene verificato che una volta chiamato il metodo sotto test il turno disdetto dal cane selezionato torni a far parte dei turni disponibili della relativa lezione
 - viene verificato che la lezione corrispondente al turno disdetto non faccia più parte delle lezioni seguite dal cane selezionato
 - viene verificato che chiamando il metodo sotto test venga restituito **false** quando il cane selezionato ha un turno scaduto.
 - viene verificato che, se il metodo sotto test viene chiamato relativamente ad un turno appartenente all'ultima lezione di un corso, il corso corrente del cane selezionato torni ad essere quello corrispondente alla lezione prenotata.
 - viene verificato che, se il metodo sotto test viene chiamato relativamente ad un turno appartenente all'ultima lezione di un corso, l'attributo `attualmenteScritto` del cane selezionato torni ad essere **true**.
 - viene verificato che, se il metodo sotto test viene chiamato relativamente ad un turno appartenente all'ultima lezione di un corso, l'ultima lezione seguita dal cane selezionato torni ad essere quella precedente alla lezione finale del corso.
 - `recuperaLezione`
 - viene verificato che chiamando il metodo sotto test quando il turno corrente del cane selezionato è scaduto, venga restituito l'elenco dei turni disponibili dell'ultima lezione seguita dal cane.
 - viene verificato che chiamando il metodo sotto test quando il turno corrente del cane selezionato non è ancora scaduto, venga restituito **null**.
 - `selezionaTurnoRecupero`
 - viene verificato che chiamando il metodo sotto test passando come parametro un turno la cui data è sovrapposta a quella del periodo in cui il cane selezionato è in affido, venga restituito **false**.
 - viene verificato che chiamando il metodo sotto test passando come parametro un turno la cui data non è sovrapposta a quella del periodo in cui il cane selezionato è in affido, venga restituito **true**.

- viene verificato che chiamando il metodo sotto test passando come parametro un turno la cui data non è sovrapposta a quella del periodo in cui il cane selezionato è in affido, l'attributo `lezioneDaRecuperare` del cane diventi **true**.
- viene verificato che chiamando il metodo sotto test passando come parametro un turno la cui data non è sovrapposta a quella del periodo in cui il cane selezionato è in affido, il turno scelto non appartenga più all'elenco dei turni disponibili della lezione da recuperare.
- viene verificato che chiamando il metodo sotto test passando come parametro un turno la cui data non è sovrapposta a quella del periodo in cui il cane selezionato è in affido, il turno corrente del cane diventi uguale al nuovo turno scelto.
- `confermaConclusioneAffido`
 - viene verificato che quando il cliente ha in affido un solo cane nel periodo selezionato e ancora manca almeno una settimana alla fine di tale periodo, chiamando il metodo sotto test viene restituito, come importo dovuto, il costo del periodo scontato del 20%.
 - viene verificato che quando il cliente ha in affido più di un cane nel periodo selezionato e manca meno di una settimana alla fine di tale periodo, chiamando il metodo sotto test viene restituito, come importo dovuto, il costo del periodo scontato del 20%.
 - viene verificato che quando il cliente ha in affido un solo cane nel periodo selezionato e manca meno di una settimana alla fine di tale periodo, chiamando il metodo sotto test viene restituito, come importo dovuto, il costo del periodo senza alcuno sconto.
 - viene verificato che quando il cliente ha in affido più di un cane nel periodo selezionato e ancora manca almeno una settimana alla fine di tale periodo, chiamando il metodo sotto test viene restituito, come importo dovuto, il costo del periodo scontato del 40% (20% + 20%).
- `confermaTimbroTurno`
 - viene verificato che quando: il cliente ha un solo cane iscritto ad un determinato corso, il turno corrente del cane non corrisponde alla prima lezione del corso ma corrisponde ad una lezione da recuperare; chiamando il metodo sotto test venga restituito, come importo dovuto, il 50% del costo del corso diviso il numero di lezioni previste da quest'ultimo.
 - viene verificato che quando: il cliente ha più di un cane iscritto ad un determinato corso, il turno corrente del cane non corrisponde ad una lezione da recuperare ma alla prima lezione del corso; chiamando il metodo sotto test venga restituito, come importo dovuto, il costo del corso scontato del 30%.
 - viene verificato che quando: il cliente ha un solo cane iscritto ad un determinato corso, il turno corrente del cane non corrisponde ad una lezione da recuperare ma alla prima lezione del corso; chiamando il metodo sotto test venga restituito, come importo dovuto, il costo del corso senza alcuno sconto.
 - viene verificato che quando: il cliente ha più di un cane iscritto ad un determinato corso, il turno corrente del cane corrisponde alla prima lezione del corso e allo stesso tempo ad una lezione da recuperare; chiamando il metodo sotto test venga restituito, come importo dovuto, il 50% del costo del corso diviso il numero di lezioni previste da quest'ultimo più il costo del corso scontato del 30%.
 - viene verificato che quando: il cliente ha solo un cane iscritto ad un determinato corso, il turno corrente del cane corrisponde alla prima lezione del corso e allo stesso tempo ad una lezione da recuperare; chiamando il metodo sotto test venga restituito, come importo dovuto, il 50% del costo del corso diviso il numero di lezioni previste da quest'ultimo più il costo del corso senza alcuno sconto.

- Cane
 - verificaldoneitaTurno
 - viene verificato che chiamando il metodo sotto test quando il turno corrente del cane è uguale ad un turno valido, venga restituito **true**.
 - viene verificato che chiamando il metodo sotto test quando il turno corrente del cane è uguale ad un turno non valido, venga restituito **false**.
 - viene verificato che chiamando il metodo sotto test quando il turno corrente del cane è uguale **null**, venga restituito **false**.