

Sommario

2.1            Introduzione..... 2

    2.2.1.        Modello di Dominio ..... 2

    2.2.2.        Diagramma di sequenza di sistema..... 4

    2.2.3.        Contratti delle operazioni ..... 5

    2.3.1        Diagrammi di Sequenza..... 7

    2.3.2        Diagrammi delle classi ..... 10

## 2. Elaborazione – Iterazione 3

### 2.1 Introduzione

La fase di elaborazione ha l'obiettivo di: raffinare la Visione, implementare il nucleo dell'architettura del software, risolvere le eventuali problematiche collegate ai rischi, individuare i requisiti e fornire stime realistiche del piano di lavoro. In questa fase si analizzano e si implementano i seguenti casi d'uso:

*UC7: Gestione Report*, è un caso d'uso in cui il sistema deve generare report sulle vendite (piatti più ordinati, ricavi giornalieri/settimanali).

*UC9: Feedback clienti*, con questo caso d'uso i clienti devono poter lasciare valutazioni e commenti dopo il pasto.

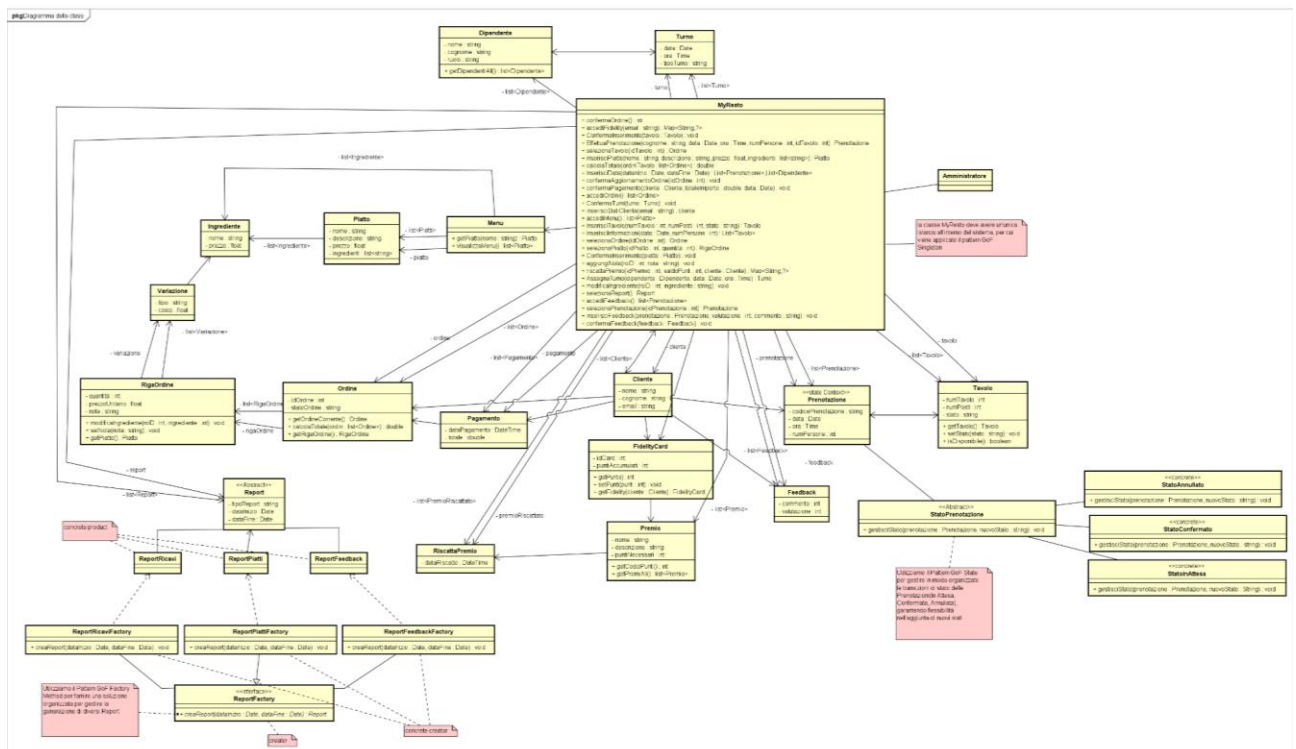
### 2.2 Analisi Orientata agli Oggetti

L'analisi orientata agli oggetti è basata sulla descrizione del dominio pensata ad oggetti. Gli strumenti utilizzati sono: Modello di Dominio, SSD (Sequence System Diagram) e Contratti delle operazioni.

#### 2.2.1. Modello di Dominio

La Modellazione del Business comprende la stesura del Modello di Dominio ed un elaborato grafico che identifica i concetti, gli attributi e le associazioni significative. Dopo una valutazione dello scenario principale di successo dei casi d'uso UC7 e UC9 sono state identificate le seguenti classi concettuali:

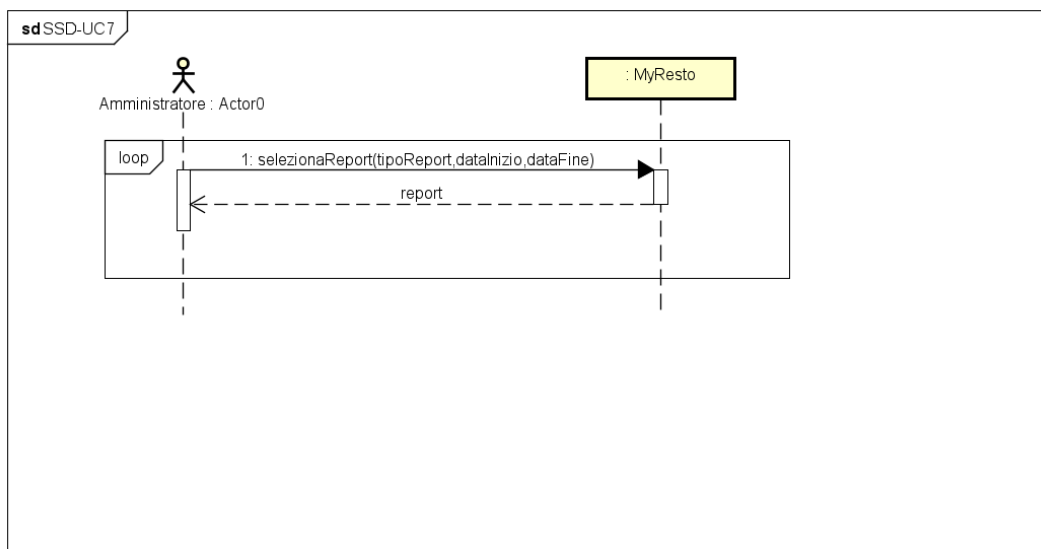
- Cliente: È sia attore che entità nel modello. Rappresenta l'utente finale del ristorante, interagisce con il sistema (come attore) per effettuare prenotazioni e altre operazioni.
- MyResto: È il sistema informatico che gestisce tutte le operazioni del ristorante.
- Amministratore: Colui che interagisce con il sistema.
- Feedback: Entità che rappresenta il giudizio lasciato dal cliente dopo un'esperienza.
- Report: Entità generata dal sistema su richiesta dell'amministratore.
- Ordine: Rappresenta l'insieme dei piatti e bevande richiesti da un cliente a un tavolo in un determinato momento.
- RigaOrdine: Rappresenta una singola voce all'interno di un Ordine, specificando la quantità di un piatto e il suo prezzo al momento dell'ordine.
- Piatto: Voce del menù. Include nome, descrizione, prezzo, ingredienti, allergeni, categoria.
- Menù: contiene l'elenco dei piatti e bevande disponibili.
- Pagamento: Entità che rappresenta il pagamento di un ordine, incluso importo, metodo, data e stato.
- Ingrediente: Rappresenta un singolo componente alimentare utilizzato per comporre un Piatto.
- Variazione: Rappresenta una modifica o un'aggiunta specifica a un ingrediente o a un piatto (es. "senza glutine", "doppio formaggio").
- Turno: Indica un intervallo temporale di lavoro assegnato ai dipendenti; include data e fascia oraria.
- Dipendente: Rappresenta un membro dello staff del ristorante (es. cameriere, cuoco). Include nome, ruolo, disponibilità e i turni assegnati.
- RiscattaPremio: Registra l'evento in cui un cliente riscatta un premio utilizzando i suoi punti fedeltà.
- FidelityCard: Rappresenta la tessera fedeltà associata a un cliente, che accumula punti in base agli acquisti e permette il riscatto di premi.
- Premio: Rappresenta un regalo che i clienti possono riscattare utilizzando i punti della loro Fidelity Card.
- Prenotazione: Rappresenta la richiesta di un cliente per riservare un tavolo in una specifica data e orario, indicando anche il numero di persone.
- Tavolo: Rappresenta un tavolo fisico nel ristorante. Include informazioni come numero identificativo, numero massimo di posti, posizione (intero o esterno) e stato (libero, occupato, prenotato).



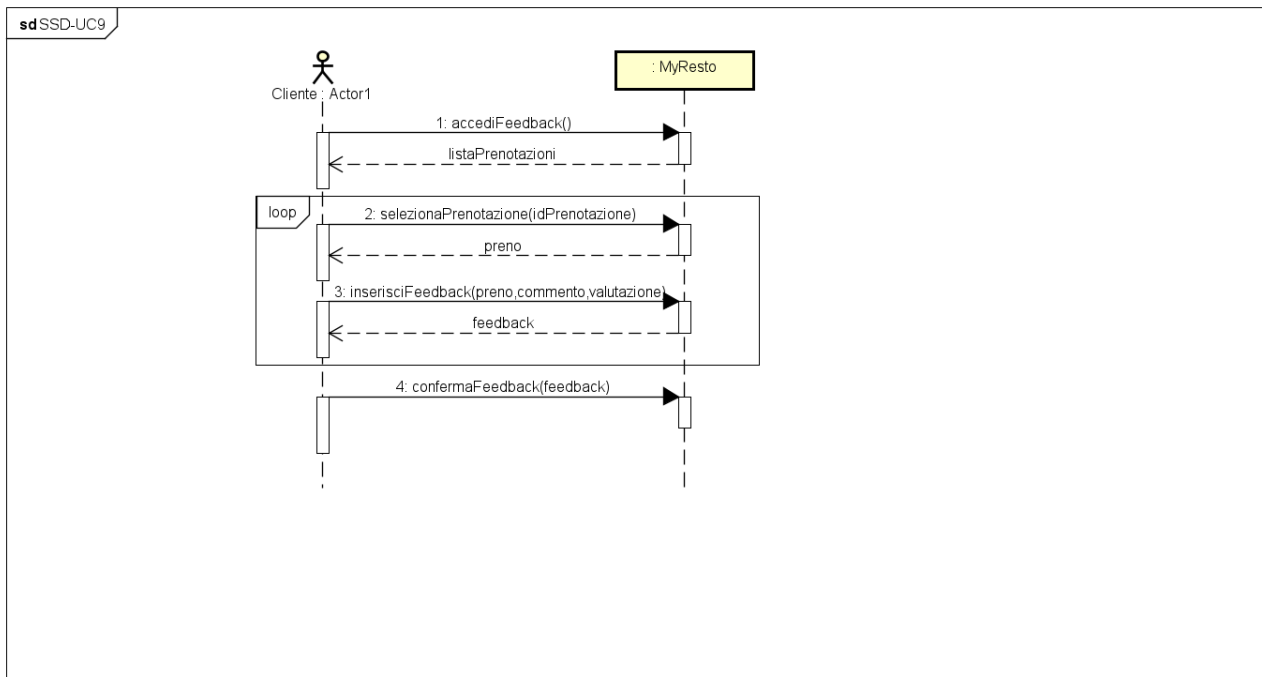
## 2.2.2. Diagramma di sequenza di sistema

Procedendo con l'analisi Orientata agli Oggetti, il passo successivo è la creazione del Diagramma di Sequenza di Sistema (SSD) al fine di illustrare il corso degli eventi di input e di output per lo scenario principale di successo nei casi d'uso scelti (UC7 e UC9), quindi avremo:

### UC7:



## UC9:



### 2.2.3. Contratti delle operazioni

Vengono ora descritte attraverso i Contratti le principali operazioni di sistema che si occupano di gestire gli eventi di sistema individuati nell'SSD.

#### UC7:

##### Contratto CO1: selezionaReport

**Operazione:** *selezionaReport(tipoReport: String, dataInizio: Date, dataFine:Date).*

**Riferimenti:** caso d'uso: Generazione Report.

**Pre-condizioni:** l'amministratore ha effettuato l'accesso alla sezione report.

##### **Post-condizioni:**

- Il sistema recupera i dati pertinenti all'intervallo di date e al tipoReport richiesto.
- Se tipoReport è "Feedback Clienti": vengono recuperati i Feedback dalla collezione Feedbacks:list<Feedback> nel periodo specificato. Il sistema calcola la media delle valutazioni dei feedback recuperati.
- Se tipoReport è "Piatti più ordinati": vengono recuperati gli Ordini dalla collezione ordiniSalvati:list<Ordine> nel periodo specificato. Il sistema conta le occorrenze di ogni Piatto all'interno degli ordini recuperati.

- Se tipoReport è "Ricavi": vengono recuperati i Pagamenti dalla collezione Pagamenti:list<Pagamento> nel periodo specificato. Il sistema calcola il totale dei ricavi sommando gli importi dei pagamenti recuperati.
- Una nuova istanza r di Report è stata creata con i dati elaborati.
- L'istanza r di Report è stata aggiunta alla collezione reportSalvati:list<Report>.
- Il file di testo reports.txt contiene le informazioni del report richiesto.

### **UC9:**

#### **Contratto CO1: accediFeedback**

**Operazione:** *accediFeedback()*

**Riferimenti:** caso d'uso: Feedback Clienti.

**Pre-condizioni:** Il Cliente ha effettuato almeno un ordine.

**Post-condizioni:**

- Il sistema recupera e fornisce l'elenco delle prenotazioni recenti associate al cliente (listaPrenotazioni).

#### **Contratto CO2: selezionaPrenotazione**

**Operazione:** *selezionaPrenotazione(idPrenotazione: int)*

**Riferimenti:** caso d'uso: Feedback Clienti.

**Pre-condizioni:** Il sistema ha precedentemente mostrato al cliente una lista di prenotazioni recenti.

**Post-condizioni:**

- Il sistema cerca la prenotazione specificata da idPrenotazione all'interno della collezione Prenotazioni:list<Prenotazione>.
- L'istanza della prenotazione (preno) corrispondente a idPrenotazione viene recuperata.

#### **Contratto CO3: inserisciFeedback**

**Operazione:** *inserisciFeedback(preno:Prenotazione,commento:String, valutazione:int).*

**Riferimenti:** caso d'uso: Feedback Clienti.

**Pre-condizioni:**

- Una prenotazione valida (preno) è stata selezionata dal cliente e passata all'operazione.
- Il commento e la valutazione sono stati forniti dal cliente e sono validi.

**Post-condizioni:**

- Una nuova istanza f di Feedback viene creata utilizzando i parametri preno, commento e valutazione.
- L'istanza f di Feedback viene restituita al sistema.

**Contratto CO4: confermaFeedback**

**Operazione:** *confermaFeedback(feedback: Feedback )*

**Riferimenti:** caso d'uso: Feedback Clienti.

**Pre-condizioni:** Un'istanza di Feedback valida è stata creata e i suoi dati sono stati forniti dal cliente.

**Post-condizioni:** L'istanza feedback viene aggiunta alla collezione Feedbacks:list<Feedback>.

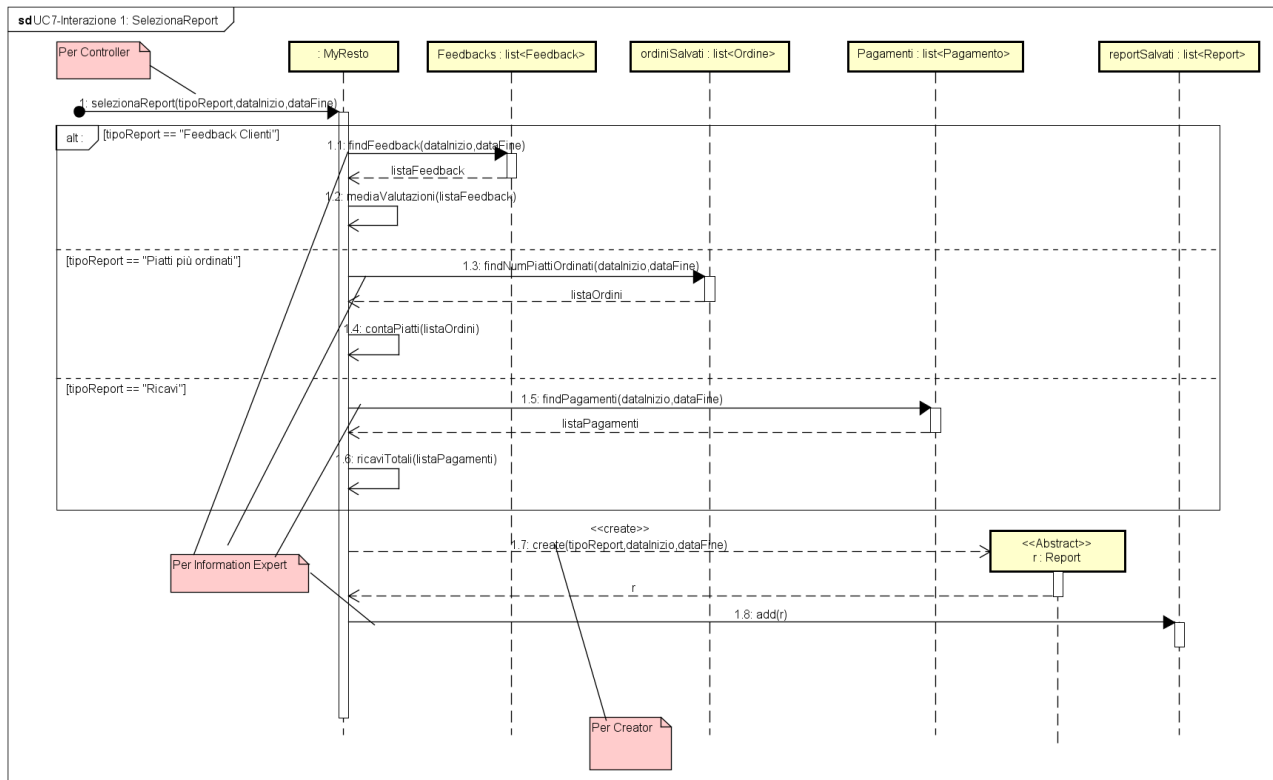
## 2.3 Progettazione

La progettazione orientata agli oggetti è la disciplina di UP interessata alla definizione degli oggetti software, delle loro responsabilità e a come questi collaborano per soddisfare i requisiti individuati nei passi precedenti. L'elaborato principale di questa fase, che è stato preso in considerazione, è il **Modello di Progetto**, ovvero l'insieme dei diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Interazione) che da un punto di vista statico (Diagrammi delle Classi). Seguono dunque i diagrammi di interazione più significativi e il diagramma delle classi relativi ai casi d'uso UC7 e UC9 determinati a seguito di un attento studio degli elaborati scritti in precedenza.

### 2.3.1 Diagrammi di Sequenza

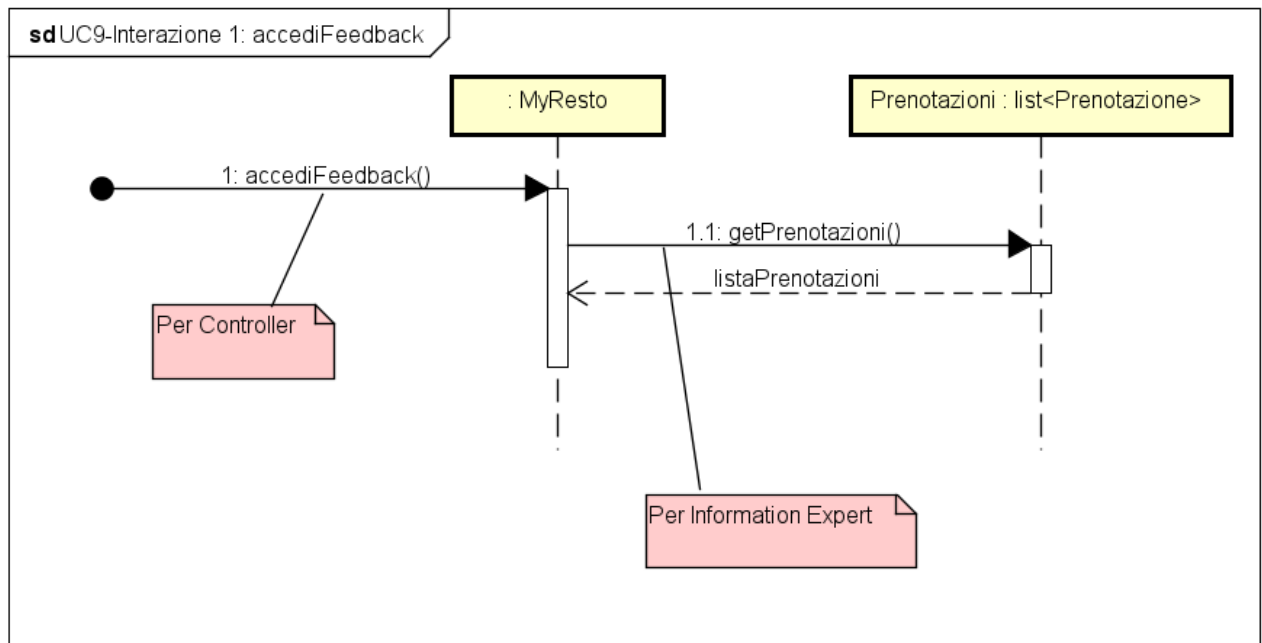
***UC7:***

- ***selezionaReport***



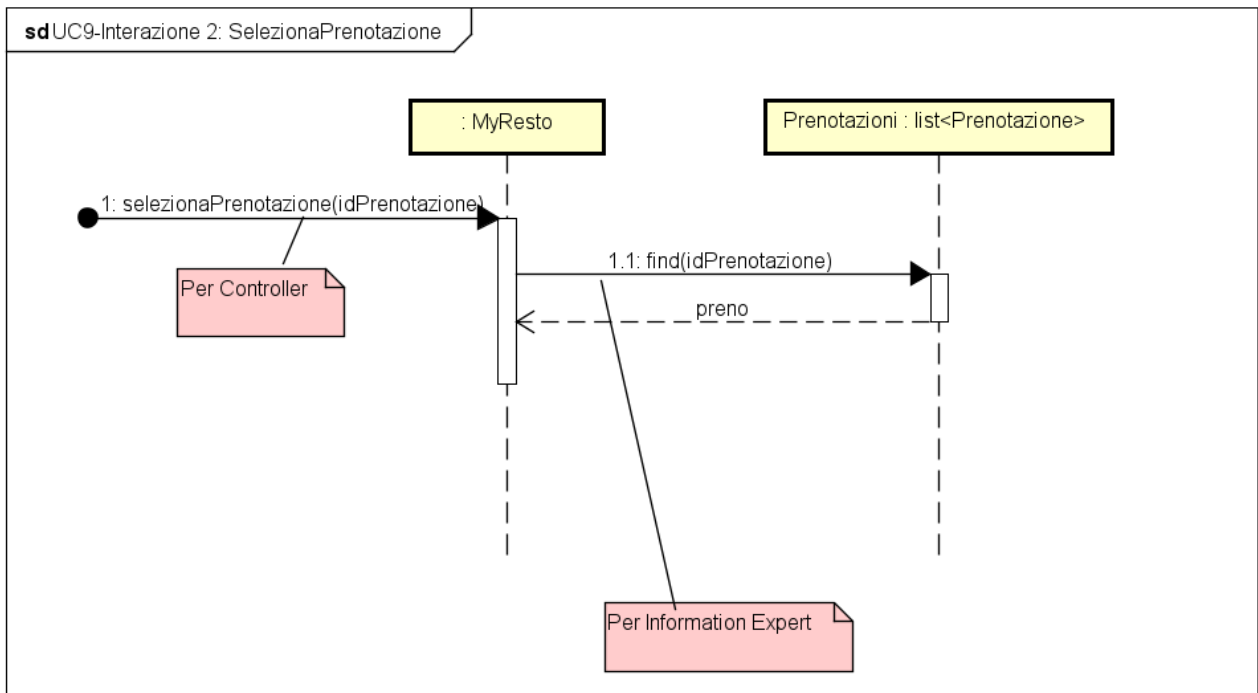
## UC9:

- accediFeedback***

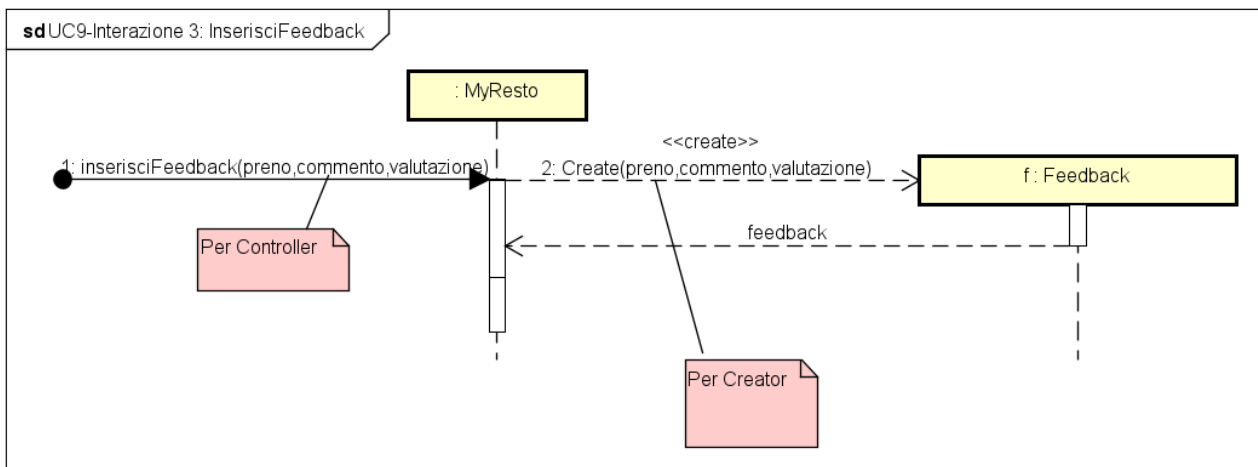


- selezionaPrenotazione***





- inserisciFeedback***



- confermaFeedback***



- Il pattern Factory Method permette di delegare la creazione di oggetti a sottoclassi specializzate, mantenendo il codice (es. il modulo di reporting) indipendente dalle classi concrete.
- Estensibilità: Aggiungere un nuovo tipo di oggetto (es. un nuovo formato di report) richiede solo l'implementazione di una nuova factory, senza alterare la logica di business.
- Coerenza con le responsabilità del sistema: Le entità dei report ad esempio condividono un'interfaccia comune, ma hanno logiche di creazione differenti. In questo modo il Factory method incapsula le differenze in classi dedicate