

Relazione Lavoro svolto Task 4 Seconda Iterazione

Gruppo G15

1. Introduzione

Durante la seconda iterazione (fase di elaborazione) il team, nel corso di otto incontri della durata media di 2 ore, per un totale di circa 16 ore complessive, ha raffinato gli artefatti prodotti durante la prima iterazione.

2. Use Case Model

Oggetto dei primi incontri è stato il raffinamento dei Casi D'Uso prodotti nella precedente iterazione.

In particolare, si è scelto di focalizzarsi sui Casi D'Uso di priorità maggiore quali Crea Partita e Crea Round, utilizzando per la specifica una formalità di tipo fully-dressed. I casi d'uso raffinati sono i seguenti:

Use Case ID 01: Crea Partita

GENERAL CHARACTERISTICS

Primary Actor	Game Engine
Secondary Actors	Nessuno
Preconditions	Nessuna
Success Post Condition	Viene creato uno Storico Partita
Failed Post Condition	Non è stato possibile creare uno Storico Partita

MAIN SUCCESS SCENARIO

Step	Action
1	Il caso d'uso inizia quando il Game Engine vuole creare uno Storico Partita
2	Il Game Engine fornisce l'ID del Giocatore
3	Il Game Engine fornisce le informazioni relative allo Storico Partita da creare
4	Il Game Engine viene notificato che la creazione è andata a buon fine
5	Al Game Engine viene restituito l'ID dello Storico Partita appena creato

EXTENSION SCENARIOS

Step	Branching Action
	*a. Il Game Engine fornisce dati di input non validi:
2a	Non è possibile creare lo Storico Partita ed il fallimento della creazione viene notificato al Game Engine;
3a	Lo Storico Partita viene creato con successo ma viene notificato al Game Engine la presenza di dati non validi;
	*b. Il Game Engine fornisce dati in input nulli:
3b	Lo Storico Partita viene creato con successo ma viene notificato al Game Engine la presenza di dati nulli;

RELATED INFORMATION

Frequency	Alta
Concurrency	Potrebbero esserci chiamate concorrenti del caso d'uso
Superordinate Use Cases	Nessuno
Open Issues	<ol style="list-style-type: none">1. In caso di partita multigiocatore viene creata un'istanza del caso d'uso per ogni giocatore partecipante o il Game Engine fornisce gli ID di tutti i giocatori alla stessa istanza?2. I Dati da salvare all'interno dello Storico Partita potrebbero variare
Due Date	08/05/2023

Use Case ID 02: Crea Round

GENERAL CHARACTERISTICS

Primary Actor	Game Engine
Secondary Actors	Nessuno
Preconditions	Deve esistere almeno uno storico partita a cui assegnare il Round
Success Post Condition	Il nuovo Round e i dati ad esso associati sono salvati nello Storico Partita
Failed Post Condition	Non è stato possibile salvare il nuovo Round nello Storico Partita

MAIN SUCCESS SCENARIO

Step	Action
1	Il caso d'uso inizia quando il Game Engine vuole registrare un Round all'interno di uno Storico Partita
2	Il Game Engine fornisce l'ID dello Storico Partita in cui registrare il Round
3	Il Game Engine fornisce l'ID del Giocatore associato allo Storico Partita
4	Il Game Engine fornisce i risultati del Giocatore e del Robot
5	Il Game Engine viene notificato che il salvataggio è andato a buon fine
6	Al Game Engine viene restituito l'ID del Round appena salvato

EXTENSION SCENARIOS

Step	Branching Action
	*a. Il Game Engine fornisce dati di input non validi:
2a	Non è possibile registrare il Round nello Storico Partita fornito ed il fallimento del salvataggio viene notificato al Game Engine;
3a	Non è possibile registrare il Round nello Storico Partita fornito ed il fallimento del salvataggio viene notificato al Game Engine;
4a	Il Round viene creato con successo ma viene notificato al Game Engine la presenza di dati non validi;
	*b. Il Game Engine fornisce dati in input nulli:
4b	Il Round viene creato con successo ma viene notificato al Game Engine la presenza di dati nulli;

RELATED INFORMATION

Frequency	Alta
Concurrency	Potrebbero esserci chiamate concorrenti del caso d'uso
Superordinate Use Cases	ID 01: Crea Partita
Open Issues	<ol style="list-style-type: none">1. In caso di partita multigiocatore viene creata un'istanza del caso d'uso per ogni giocatore partecipante o il Game Engine fornisce gli ID di tutti i giocatori alla stessa istanza?2. I Dati da salvare all'interno del Round potrebbero variare
Due Date	08/05/2023

3. Class Diagram

In seguito ad un'analisi più accurata ed approfondita a valle del confronto avuto, si è ritenuto necessario un perfezionamento del Diagramma delle Classi.

Il Diagramma di Classe raffinato è visibile in Figura 1.

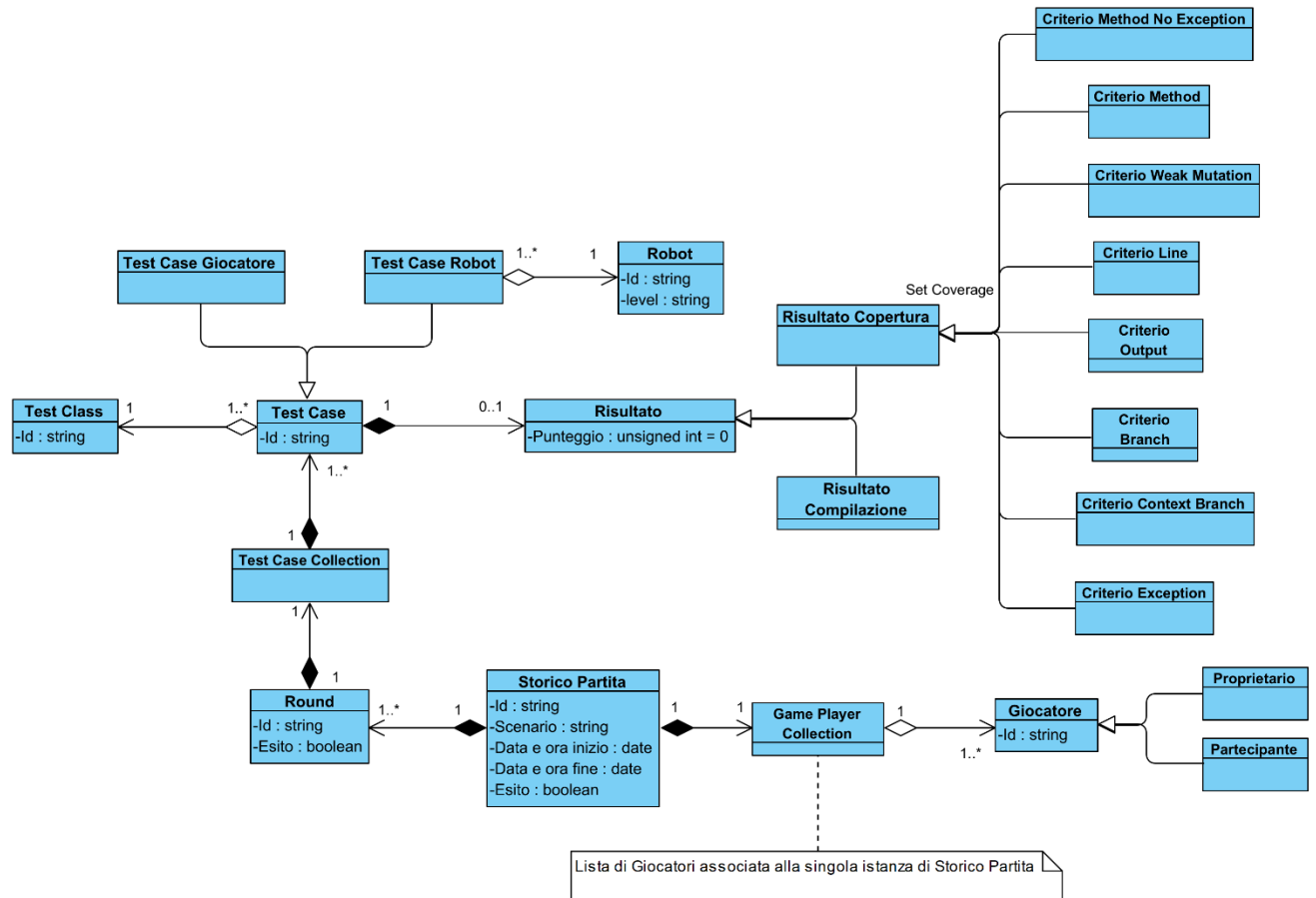


Figura 1: Class Diagram

Degna di attenzione è la modifica apportata alla Classe Lista Partecipanti, sostituita dalla classe Game Player Collection (associata ad una singola istanza di Storico Partita) contenente le diverse istanze della classe Giocatore.

4. Architecture Diagram

L'Architettura scelta per il task T4 è un'architettura di tipo Client-Server dove il Tier Client (il Game Engine) comunica con il Tier Server (la Game Repository) attraverso un API REST. In particolare, il Server è realizzato tramite un'architettura a 4 livelli CLA:

- Il Controller Layer racchiude l'implementazione dei servizi REST per la realizzazione dell'interfaccia CRUD tra Client e Server;
- L'Entity Layer associa i dati prelevati dal Database con le classi specificate dal Class Diagram affinché il Controller Layer possa manipolare i dati;
- Il DAO Layer contiene le classi che specificano i metodi per realizzare la persistenza dei dati necessari;
- Il Database Layer contiene il Database e le interfacce necessarie per stabilire una comunicazione con esso;

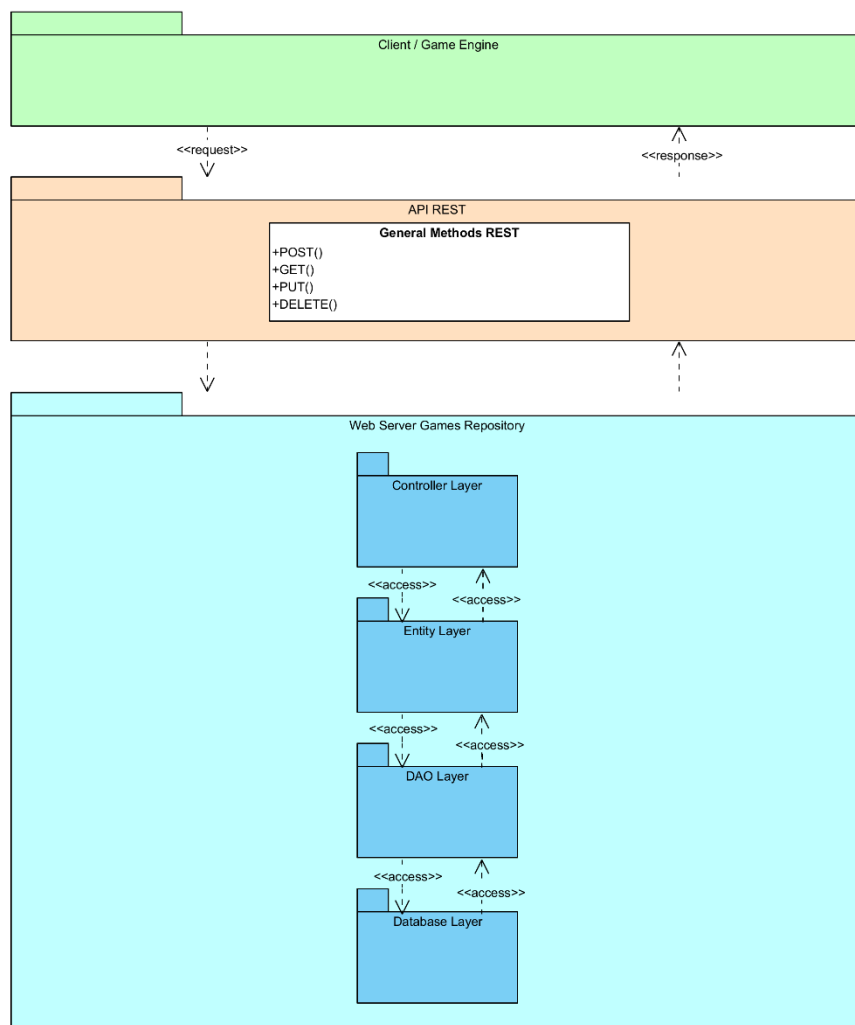


Figura 2: Architecture Diagram

5. Sequence Diagram

Partendo dalla definizione dell'architettura e dal raffinamento dei Casi D'Uso si è sviluppato una serie di Sequence Diagram UML per i Casi D'Uso Crea Partita e Crea Round.

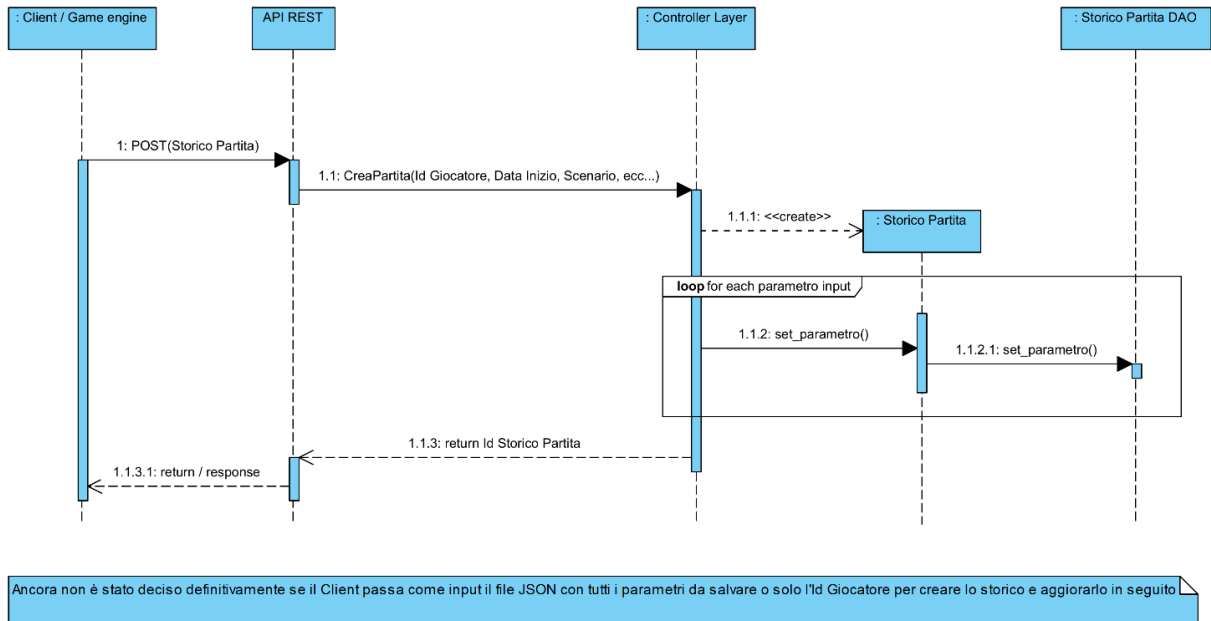


Figura 3: Sequence Diagram CreaPartita

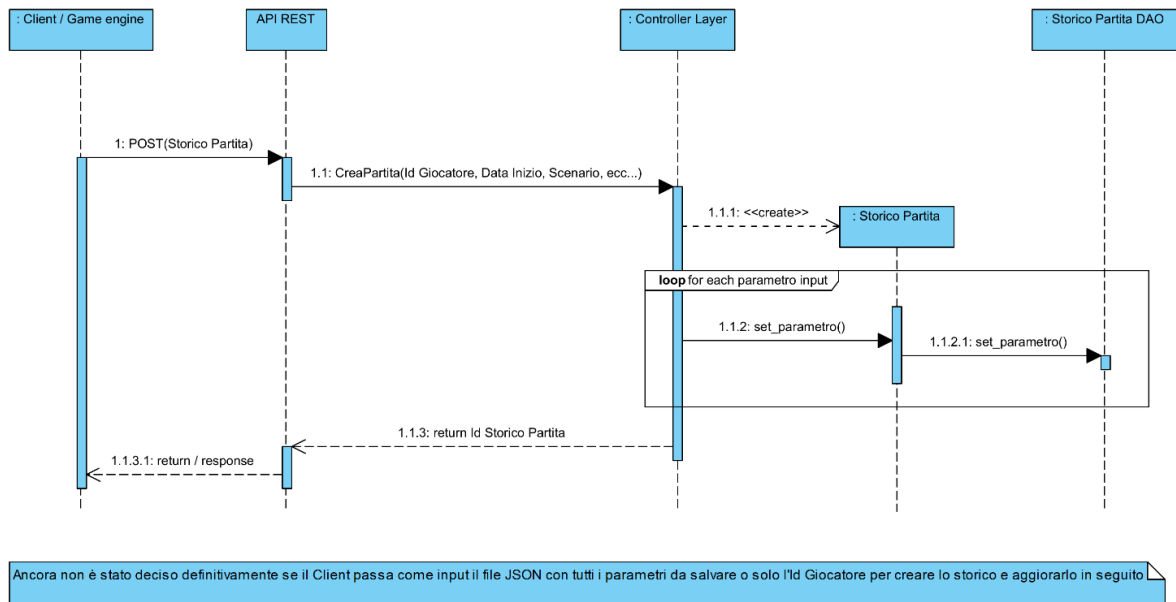


Figura 4: Sequence Diagram CreaRound

6. Tecnologie Studiate

In questa iterazione si è ritenuto necessario fare un approfondimento sulle API REST ed il framework Spring.