

---

Matteo Cartuccia  
Flavio Macciocchi

# |FIXIT|

## Documentazione di analisi

**ID:** 4-FIX-ANL-v06-r2

Data ultima modifica: 20/01/2013

Data ultima revisione: 15/01/2013

**Progetto di Ingegneria del software A.A 2012/2013**

## 1. Introduzione

Dopo aver definito e descritto i casi d'uso stiamo passando ad elaborare la fase di analisi, composta per prima cosa dall'analisi nome-verbi. Quest'ultima aiuta ad individuare le classi, gli attori le azioni e gli attributi di cui il sistema è composto.

Come secondo passo andremo a identificare le classi che compongono il sistema, con i rispettivi metodi e tipi, distinguendo per ognuna la cardinalità delle associazioni.

Studieremo poi le correlazioni tra classi individuandole all'interno dei *package*, riportando un modello di diagramma che specifichi le interazioni possibili.

Per concludere stenderemo i diagrammi di sequenza UML, questi mostrano la sequenza di messaggi tra le istanze di classe, componenti, sottosistemi o attori.

## 1.1 **Brainstorming: raccogliere informazioni**

Durante la fase di analisi il team di sviluppo si è cimentato in diversi *brainstorming*, ai quali hanno partecipato anche personaggi esterni al progetto. In questi incontri si è discusso riguardo alle classi di analisi sottoponendo al gruppo schede CRC, predisposte con il classico schema triangolare per individuare il nome delle classi, le relative responsabilità e i loro collaboratori.

L'analisi CRC è stato un modo efficace (e divertente) di coinvolgere il team e persone esterne, alle quali è stato assegnato il ruolo di utente giocatore o biochimico, o anche di finanziatore del progetto.

Questa procedura è stata inoltre portata avanti in modo parallelo all'analisi nome-verbi dei casi d'uso (capitolo 2) ed è servita ad incrementare altre parti della documentazione come quella dei requisiti o quella del glossario. Nonostante le schede CRC si siano rivelate uno strumento fondamentale per l'incremento del lavoro in generale, non riporteremo una documentazione grafica delle stesse, ma potremo comunque apprezzare il loro contributo leggendo e valutando le classi di analisi riportate nel capitolo 4 del presente documento.

## 2. Analisi nomi-verbi

Classi

Attori

Azioni

Attributi

Il sistema si occupa di gestire il ripiegamento proteico utilizzando una forma particolare di calcolo distribuito, ovvero il "*Distributed thinking*", al fine di far progredire la ricerca scientifica.

Avvalendosi di questo software sarà possibile ricavare dati attendibili e prevedere quale possa essere il ripiegamento migliore per una determinata forma proteica eliminando eventuali imperfezioni strutturali. Questo servizio si avvale del web per ricavare le informazioni necessarie alla ricerca, raccogliendo i dati in una banca dati centralizzata, accessibile ai biochimici responsabili del sistema e a particolari ingegneri informatici specializzati nella ricerca di un algoritmo ottimale finalizzato al ripiegamento proteico automatizzato.

## 2.1 Ai Giocatori

Il sistema permette al giocatore di interagire e modificare modelli tridimensionali di proteine, questi sono rappresentati da un puzzle che l'utente deve risolvere spostando in modo appropriato gli aminoacidi ed eliminando eventuali punti energetici. Ad ogni partita è associato un punteggio minimo che il giocatore deve raggiungere per passare al quadro successivo.

L'utente sarà supportato da una chat globale e una locale, da un menu che faciliti il settaggio delle impostazioni quali scelta username, cambio set colori e lingua.

Il nostro giocatore potrà visualizzare i trofei e i livelli rispettivamente ottenuti e superati, salvare le partite in qualsiasi stato, uscire e tornare a giocare in un secondo momento.

## 2.2 Ai Biochimici

Il software permette ai biochimici di introdurre nuove figure rappresentanti proteine, dandogli in oltre la possibilità di aggiungere e rimuovere anomalie sulla struttura base (legami principali) così da poter descrivere qualsiasi catena polipeptidica. Questo tipo di utente può settare a suo piacimento il punteggio minimo richiesto per il superamento del puzzle e impostare il tempo di scadenza (time to live) dello stesso. Il biochimico potrà effettuare il login con uno speciale ID e una volta entrato

nel menu potrà gestire le proteine in tutte le sue parti ovvero: struttura base, struttura aminoacido, aggiungere e rimuovere punti di energia, impostare il punteggio minimo richiesto (*score*).

## 2.3 Agli Ingegneri Informatici

Gli Ingegneri Informatici potranno collegarsi a uno speciale server SFTP per scaricare i dati in formato XML, questi sono esportati in un'apposita cartella in modo periodico dal sistema.

Inoltre questo tipo di attore potrà utilizzare un client specializzato (es. *FileZilla*) nello scambio di dati che supporti il suddetto protocollo al fine di velocizzare il download dei dati.

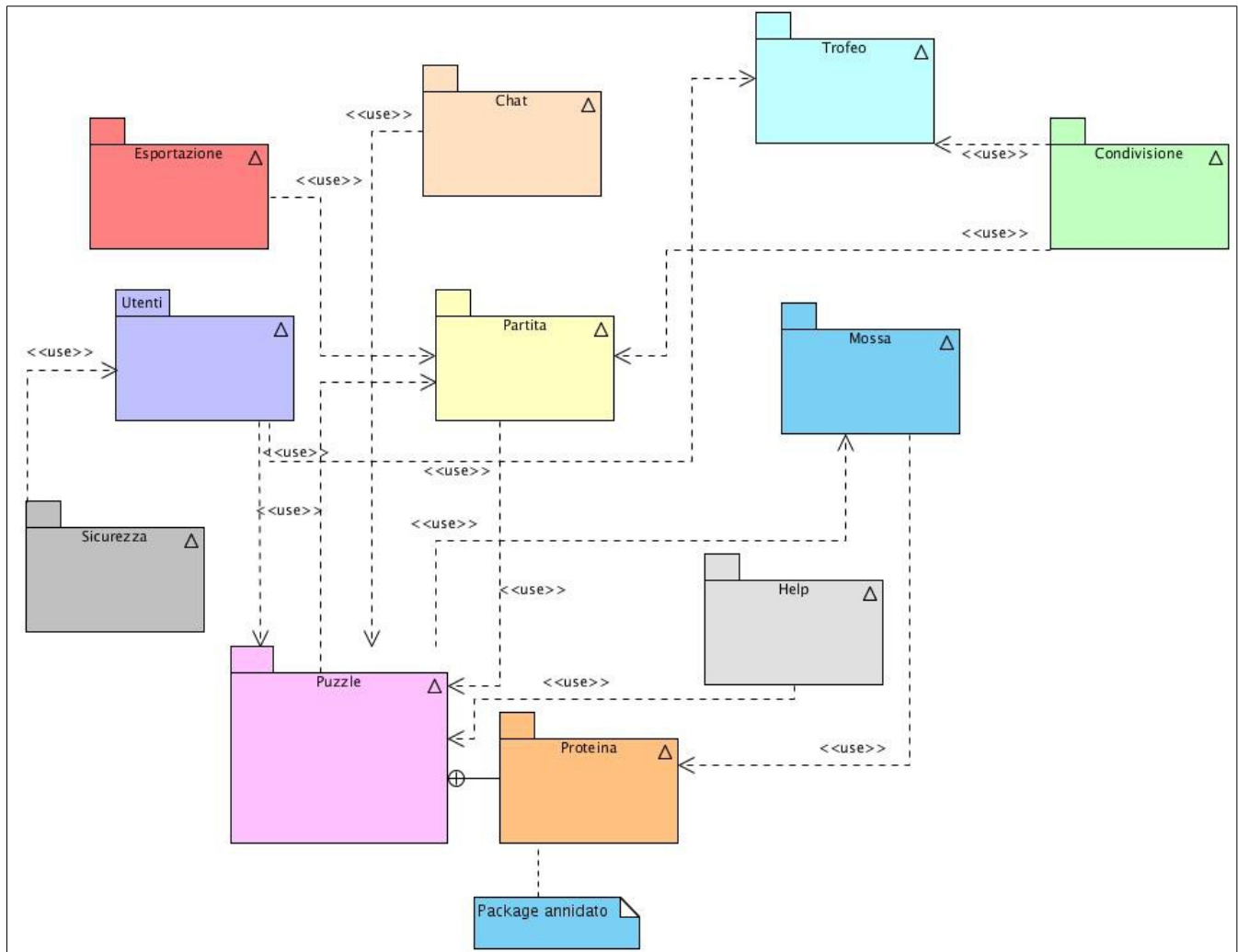
### 3. Diagramma dei *package*

Un *package* è uno speciale contenitore degli elementi del modello, ovvero un meccanismo per organizzare le classi all'interno di sottogruppi ordinati. Il seguente diagramma mostra come sono collegati tra loro i *package* contenenti le classi logicamente correlate (che forniscono servizi simili) e descritte precedentemente.

Ad ogni *package* è assegnato un *namespace* che ne identifica il contenuto, inoltre le classi raggruppate al loro interno hanno la possibilità di accedere ai metodi e alle variabili definite *protected* delle altre classi che fanno parte dello stesso *package*.

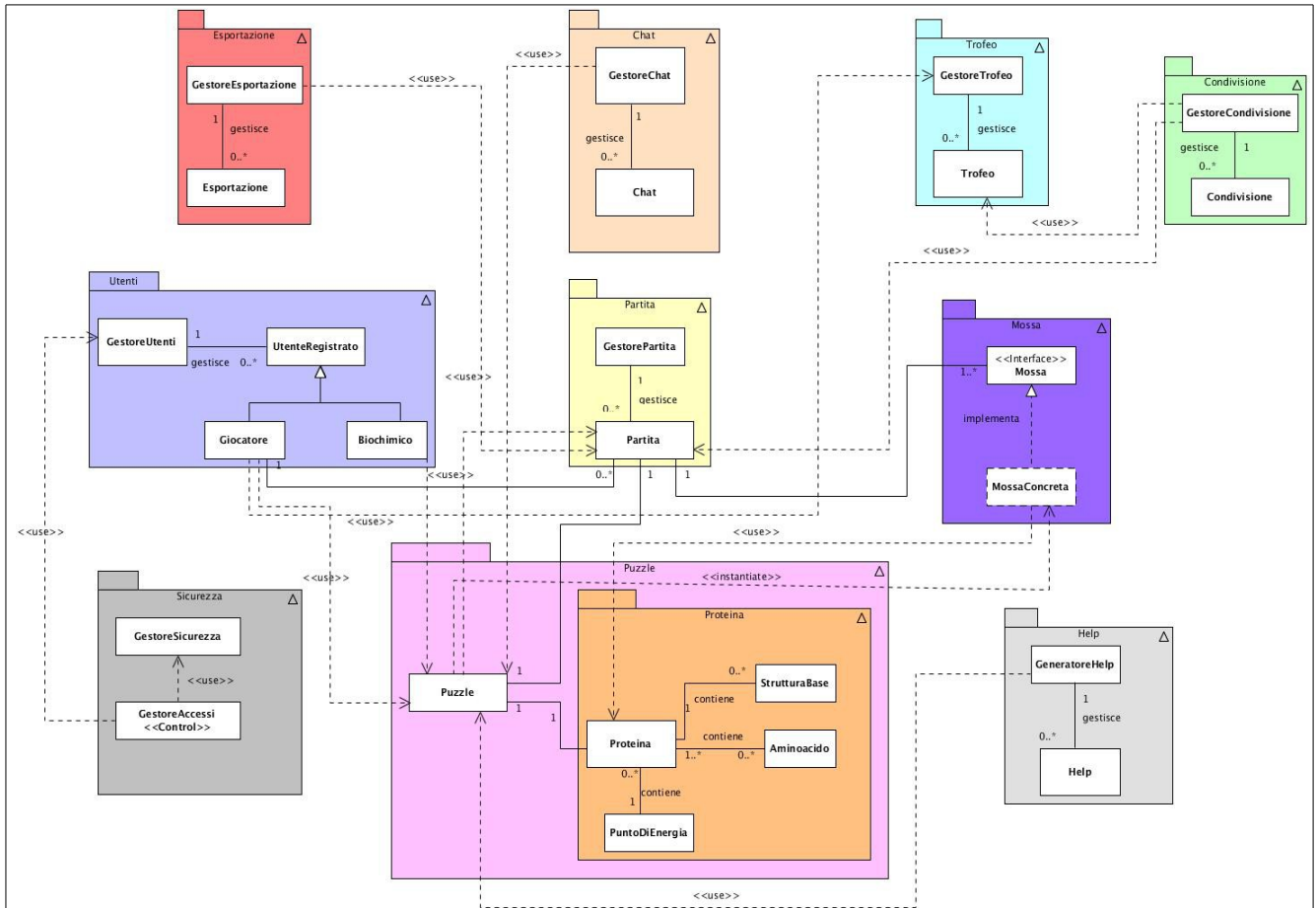
Lo scopo di questa parte dell'analisi consiste comunque nel mostrare *come* interagiscono e cooperano le classi di analisi, per definire il funzionamento dei casi d'uso esposti nel precedente documento. Ci soffermiamo per ora su un certo grado di astrazione ed approfondiremo maggiormente i contenuti nel successivo modulo, preoccupandoci di dove risiederanno le componenti fisiche del progetto stesso.

### 3.1 Modello dei package





## 3.2 Modello delle classi

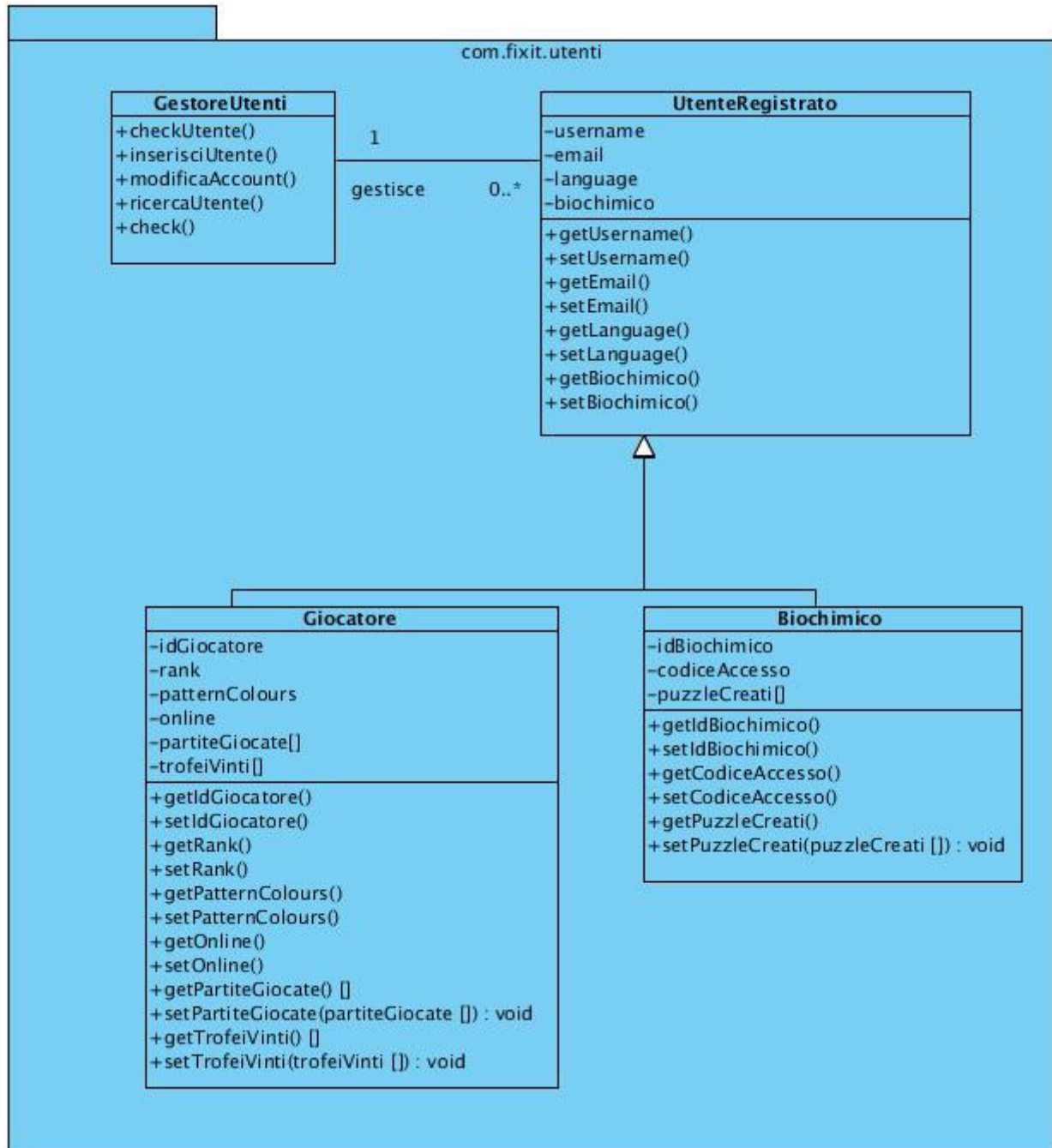


## 4. Classi di analisi

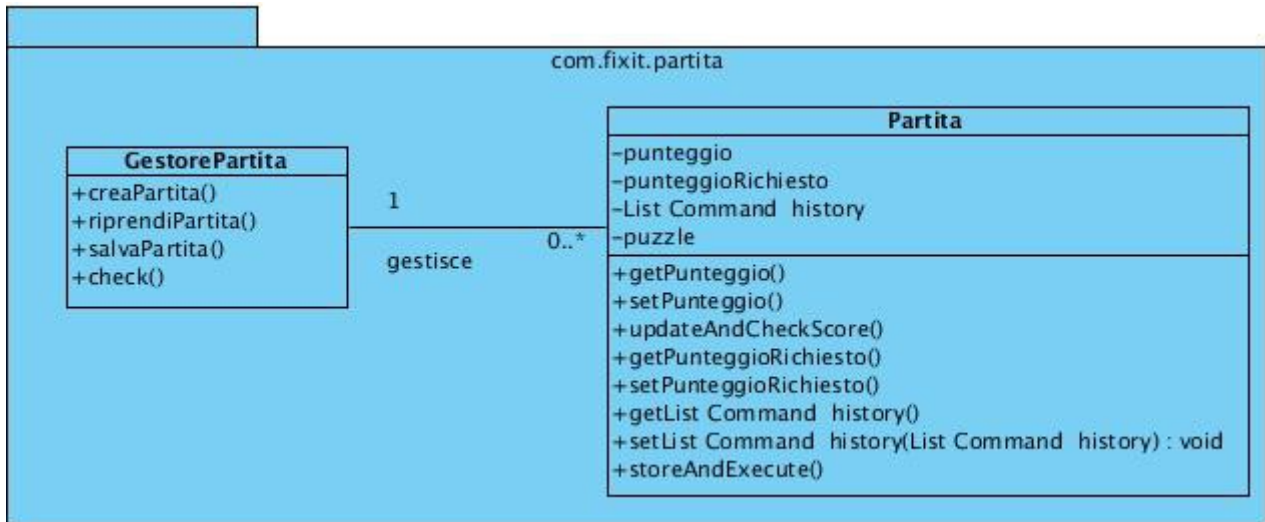
Qui di seguito riportiamo le classi di analisi utilizzando dei diagrammi appositamente studiati e riconosciuti dallo *standard UML*. Grazie alla flessibilità di quest'ultimo, è stato possibile quindi rappresentare graficamente le classi e le loro associazioni.

Con tali classi vogliamo mostrare il comportamento del nostro sistema di gioco *Fixit*, avvalendoci di una rappresentazione ad alto livello, queste poi verranno riprese in seguito per generare i cosiddetti *diagrammi di sequenza*, come indicato nell'introduzione.

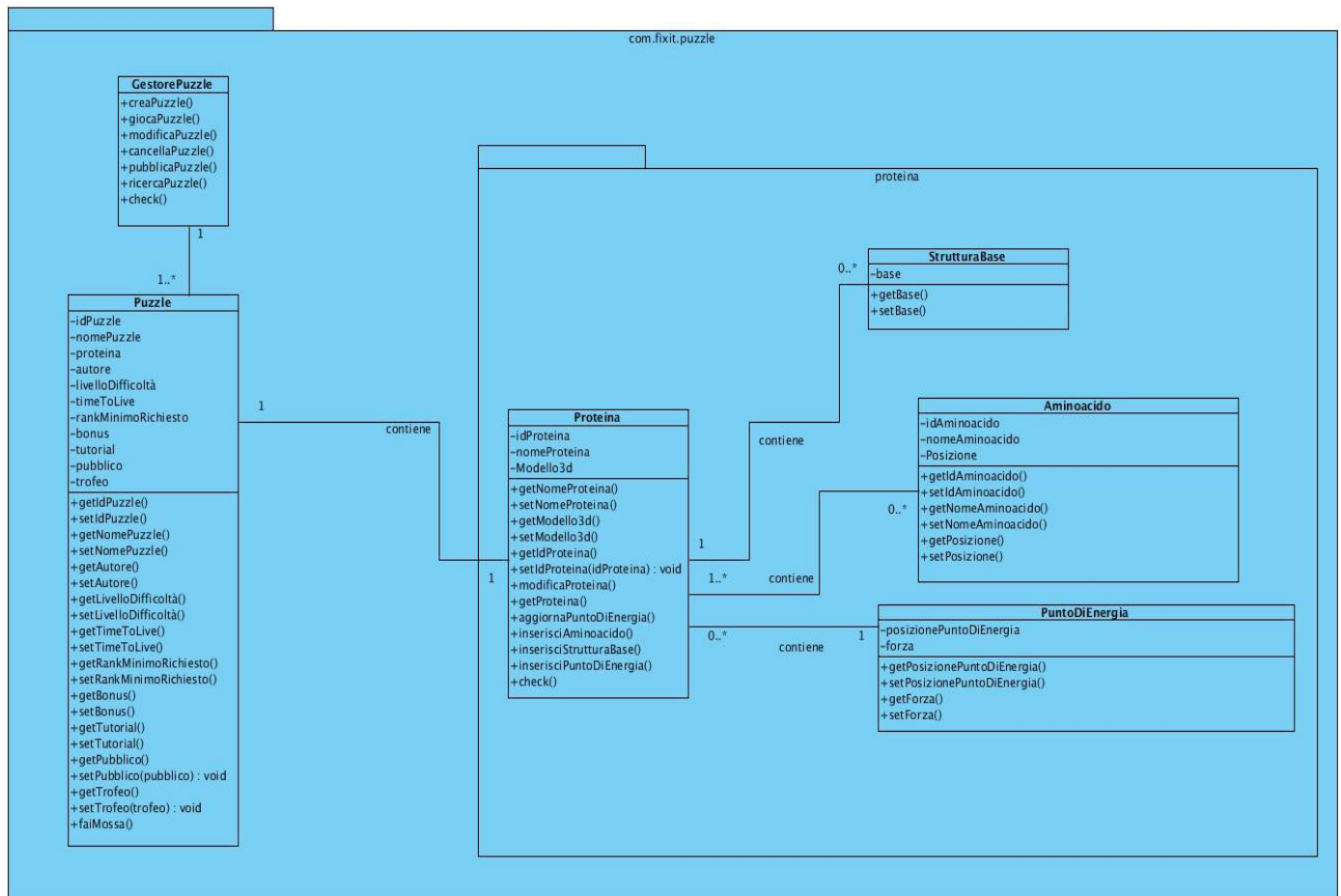
## 4.1 Package utenti



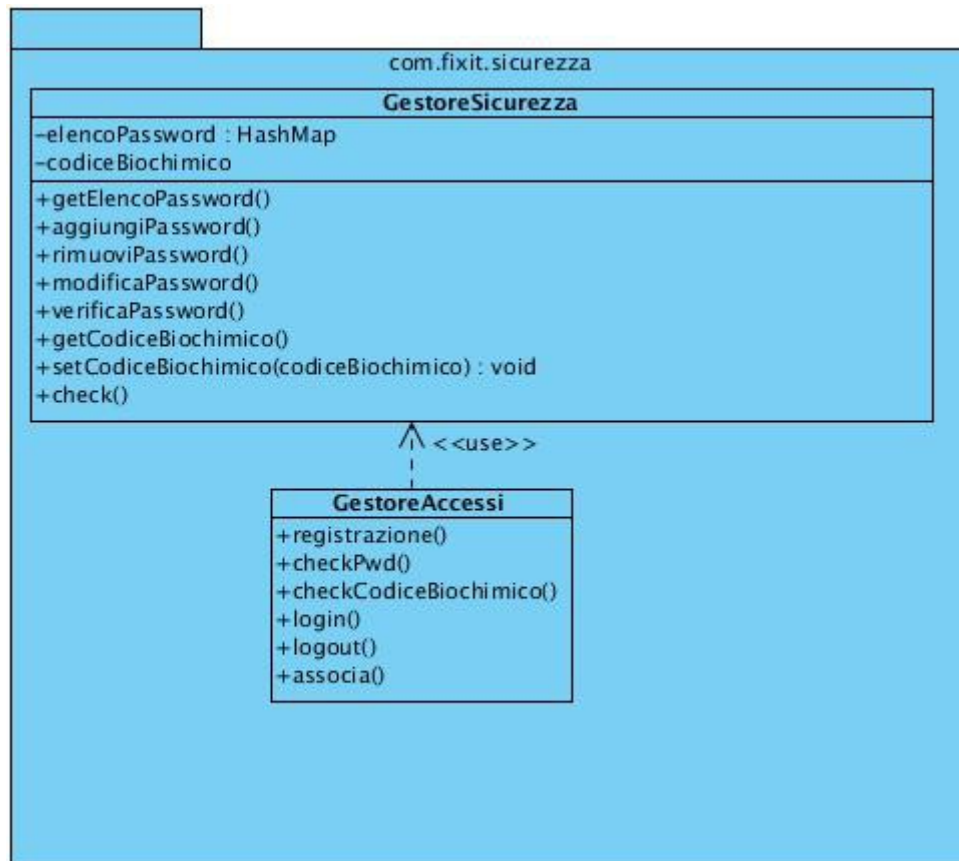
## 4.2 Package Partita



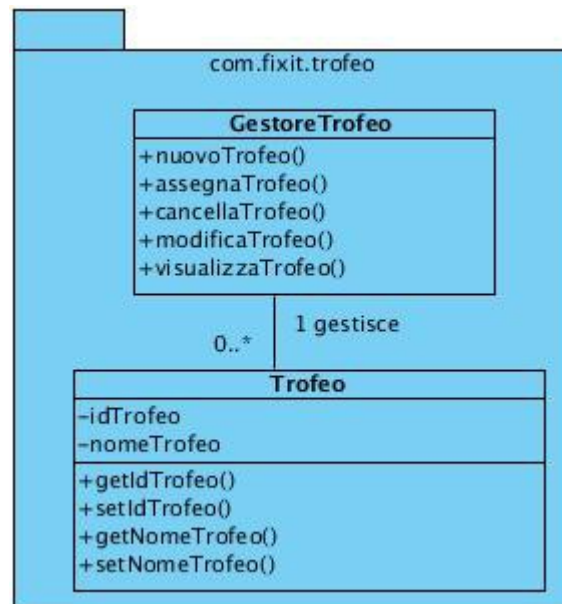
## 4.3 Package puzzle



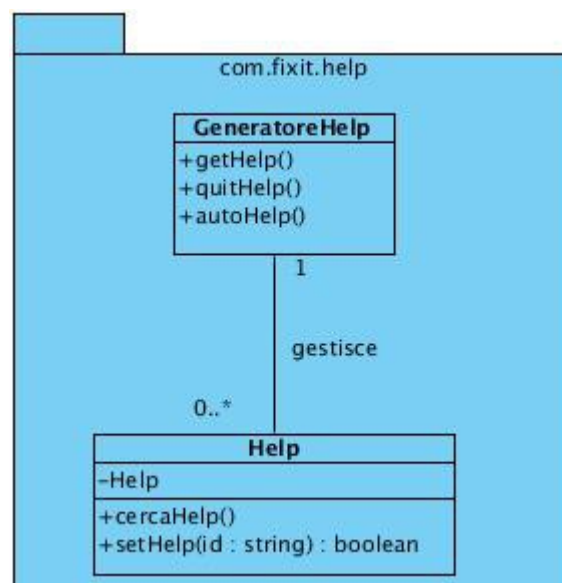
## 4.4 Package sicurezza



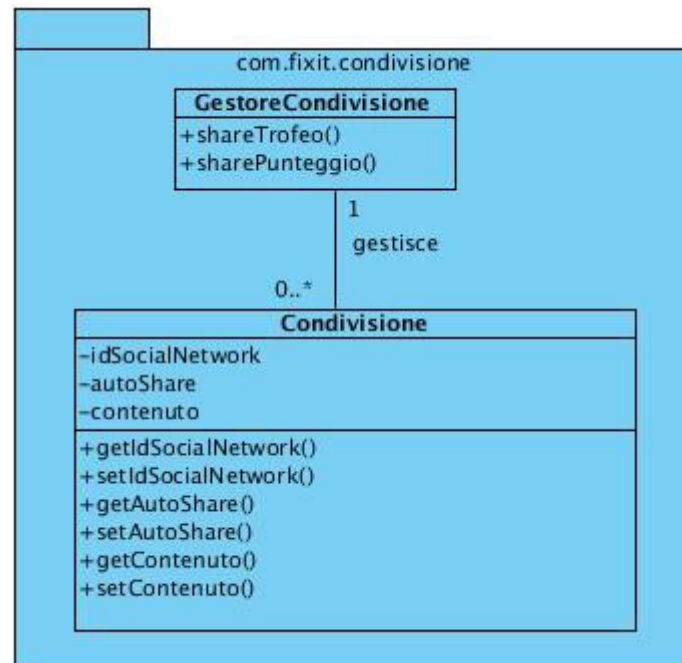
## 4.5 Package trofeo



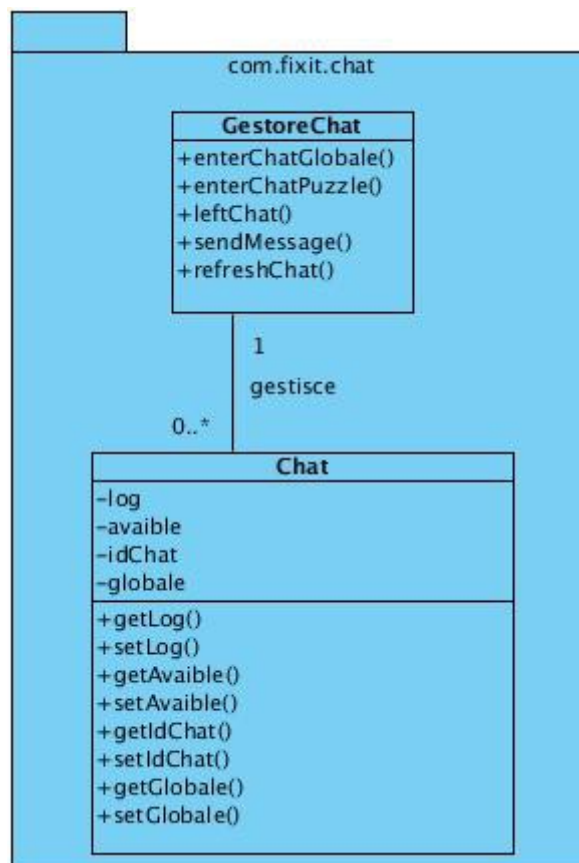
## 4.6 Package help



## 4.7 Package condivisione

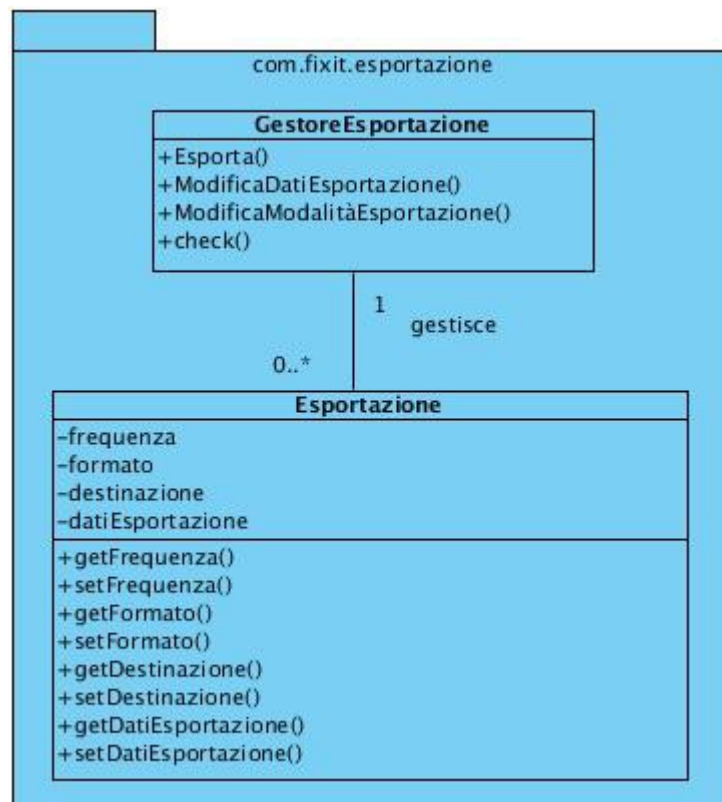


## 4.8 Package chat

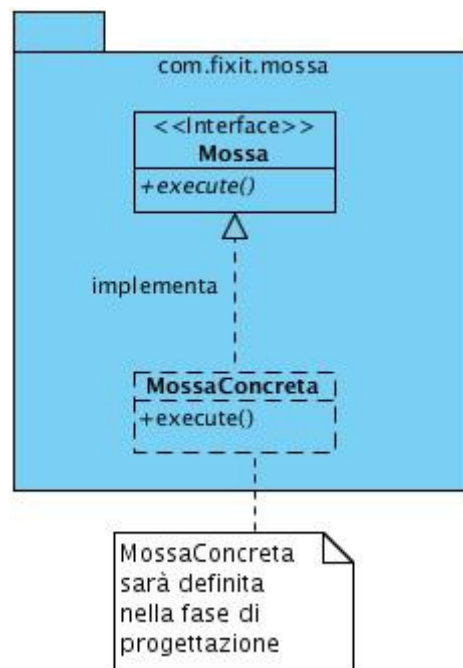




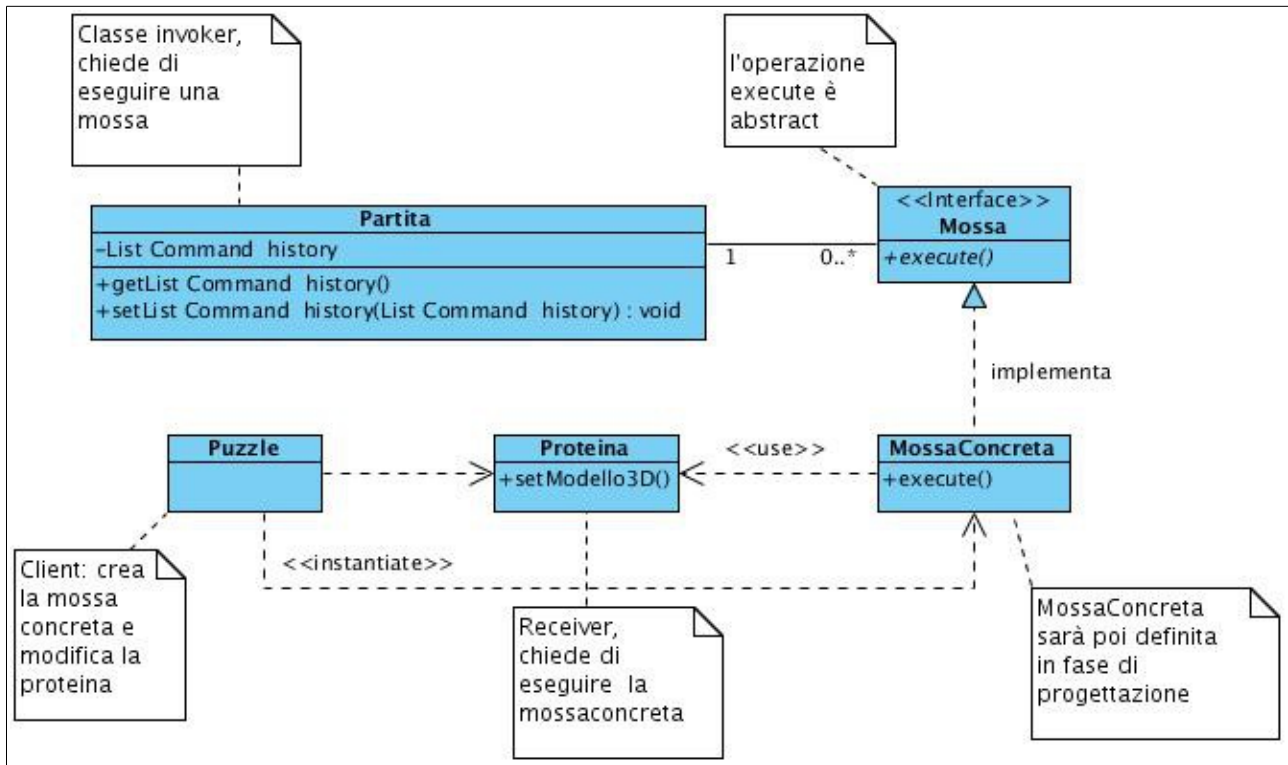
## 4.9 Package esportazione



## 4.10 Package mossa



## 4.11 Uso del *pattern*



Il *command pattern*<sup>1</sup> ci permette di isolare la mossa rendendo più agevole azioni quali l'*undo* o il *redo*. Si tratta di un *pattern* (nel nostro caso basato su linguaggio java) che viene utilizzato quando si ha la necessità di disaccoppiare l'invocazione di un comando dai suoi dettagli implementativi, separando colui che invoca il comando da colui che esegue l'operazione. Tale metodo viene realizzato attraverso questa catena: *Client*->*Invocatore*->*Ricevitore*.

<sup>1</sup> Command pattern: [http://en.wikipedia.org/wiki/Command\\_pattern](http://en.wikipedia.org/wiki/Command_pattern)

## 5. Diagrammi di sequenza

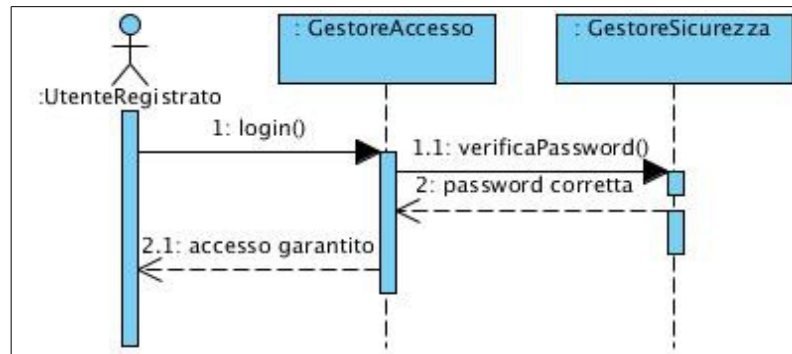
Ciò che segue sono i diagrammi di sequenza, che descrivono uno scenario tipico del nostro sistema, esponendo lo scambio di messaggi tra attori e i vari moduli appartenenti al dominio.

Nello specifico si delineano le relazioni che intercorrono tra attori, oggetti appartenenti al dominio ed entità corrispondenti al sistema preso in esame.

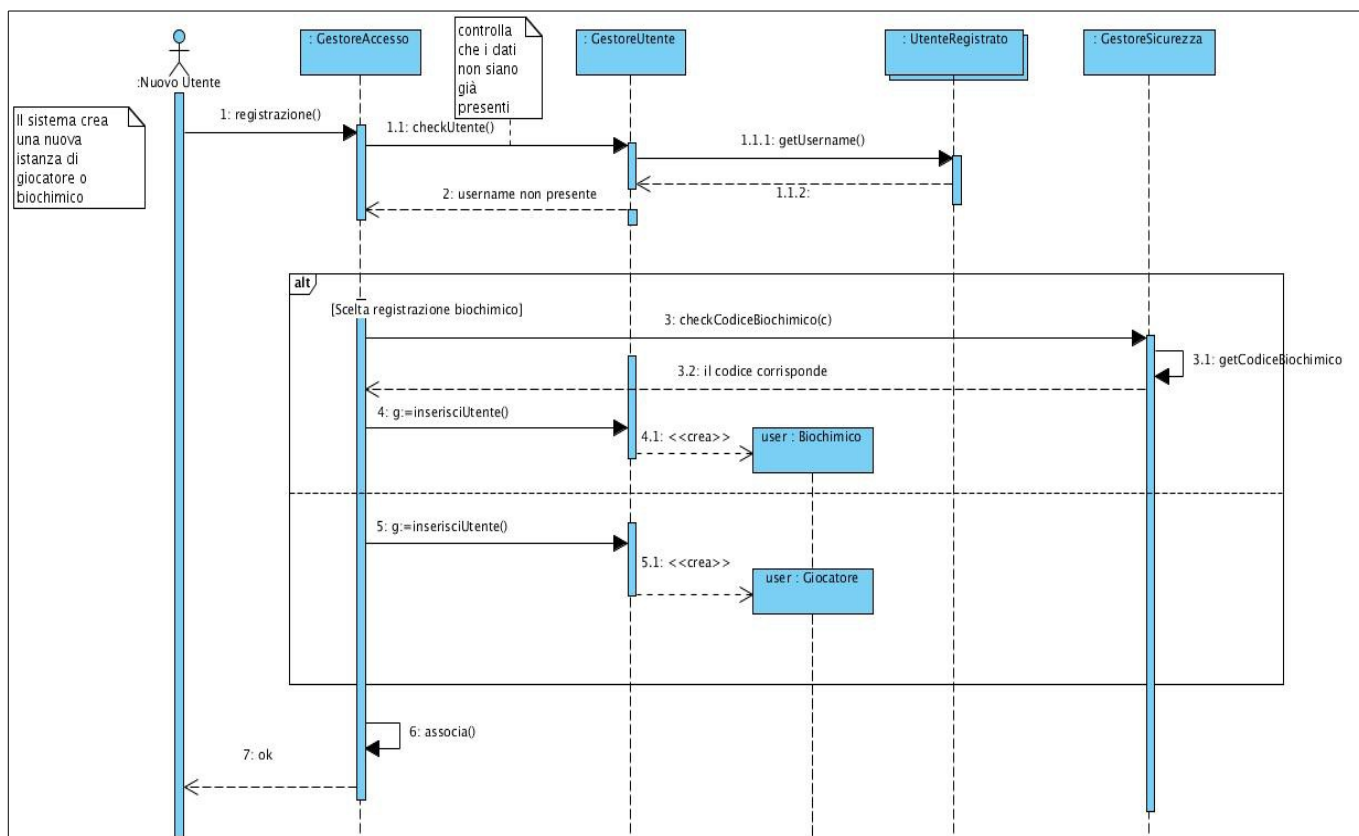
Ricordiamo che in questa fase si ha una visione d'insieme, i dettagli implementativi relativi ai diagrammi saranno riportati nella documentazione inerente alla progettazione ( doc **5-FIX-PRO-v00-r00** ).

Come vedremo i nostri diagrammi di sequenza mettono in luce lo scambio di messaggio su linee di vita, generando anche una visione temporale del sistema in correlazione con gli oggetti di dominio.

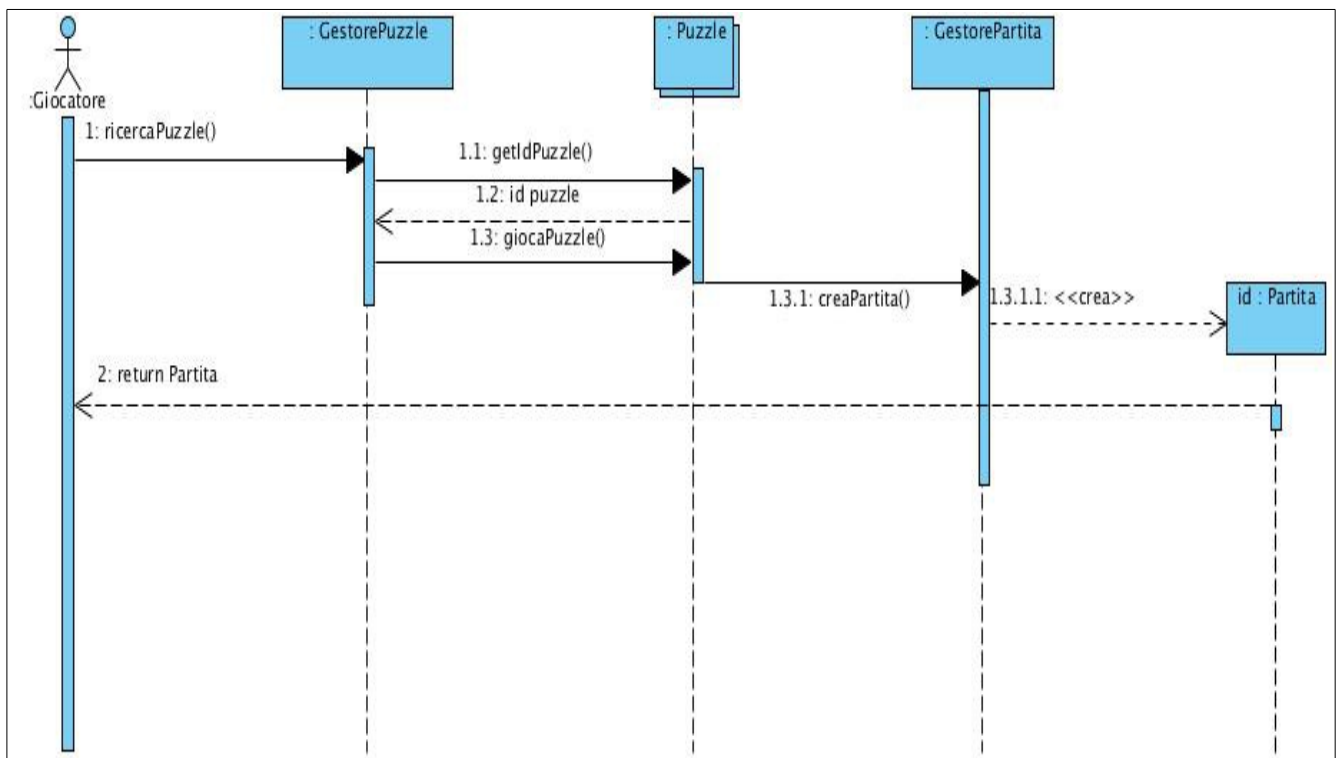
## 5.1 Login



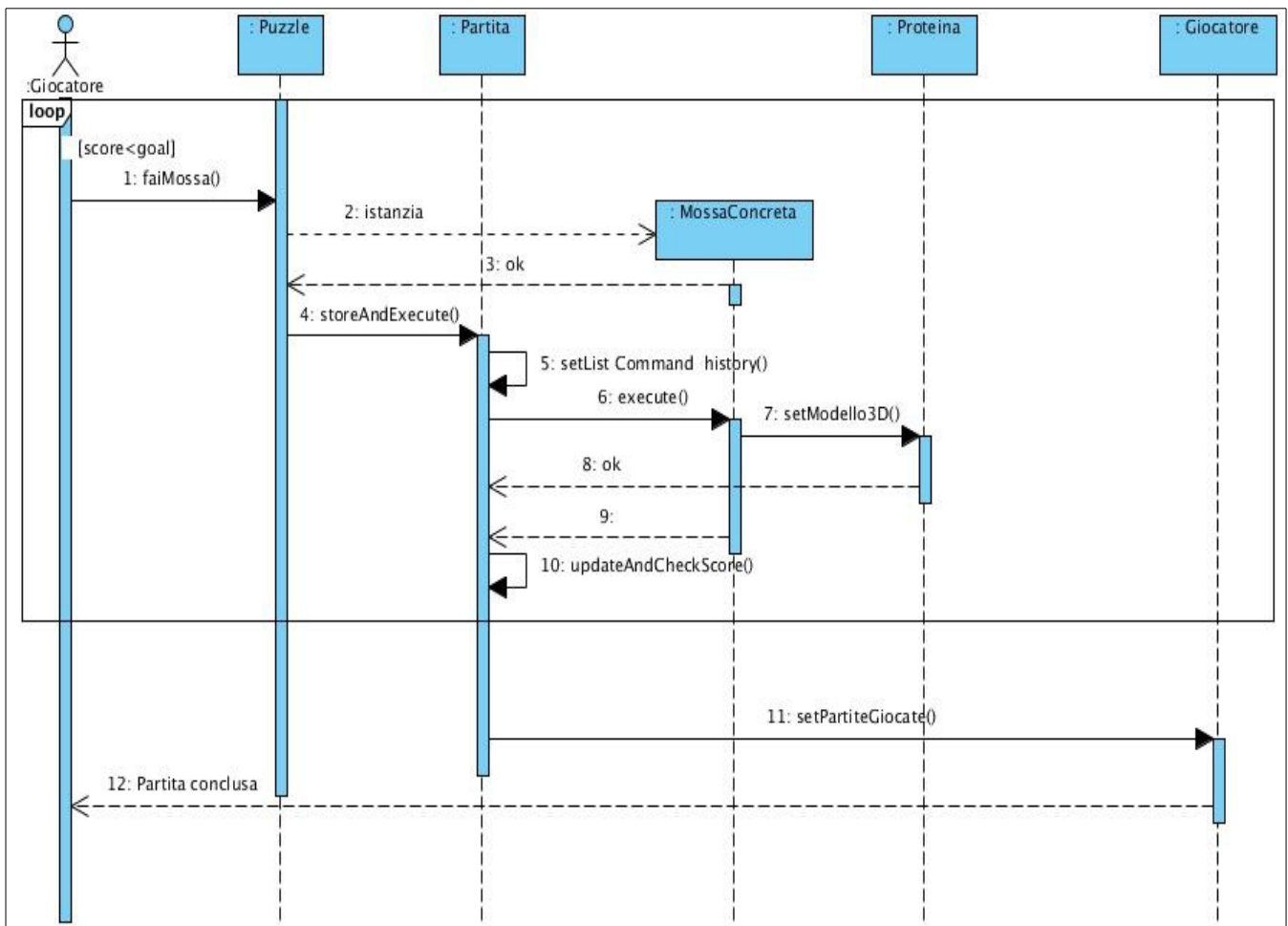
## 5.2 Crea account



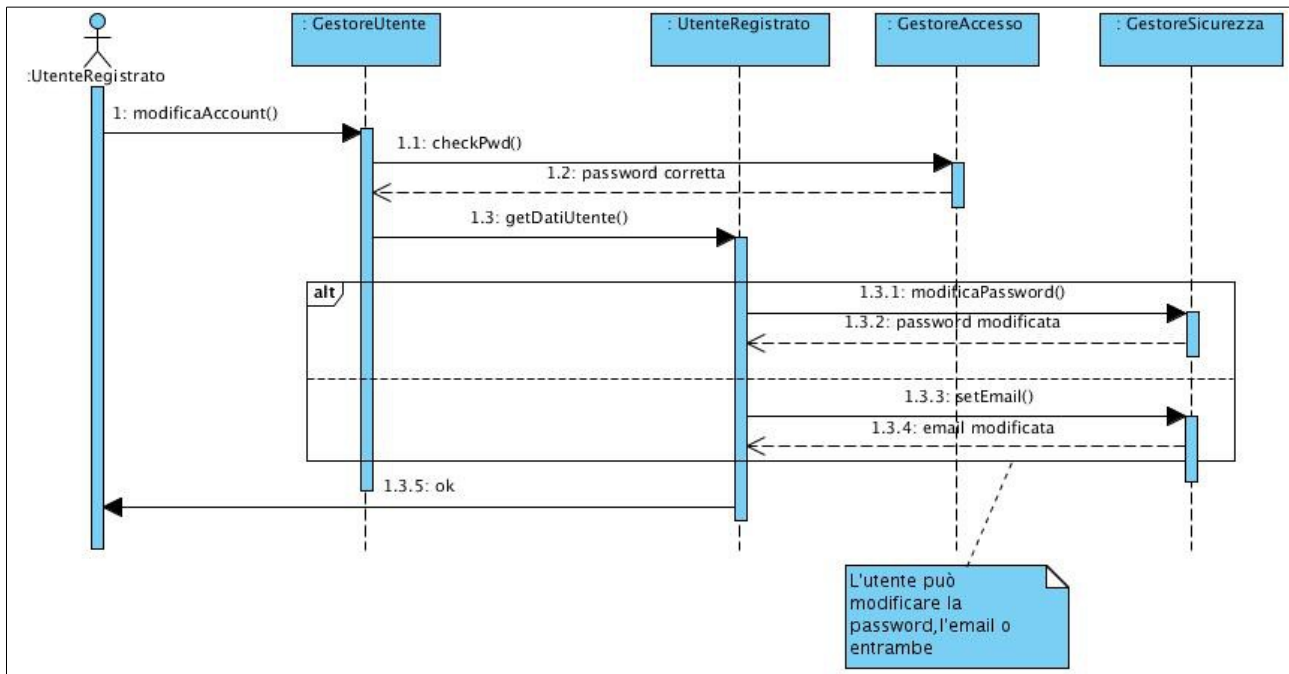
## 5.3 Scegli partita



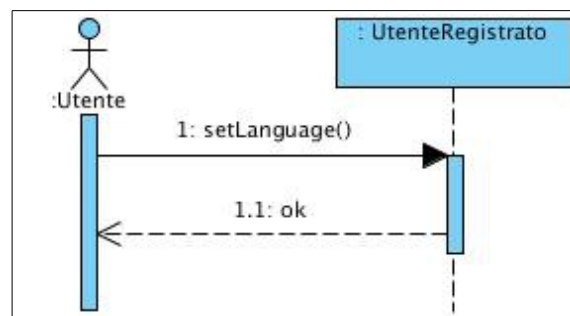
## 5.4 Gioca partita



## 5.5 Modica account

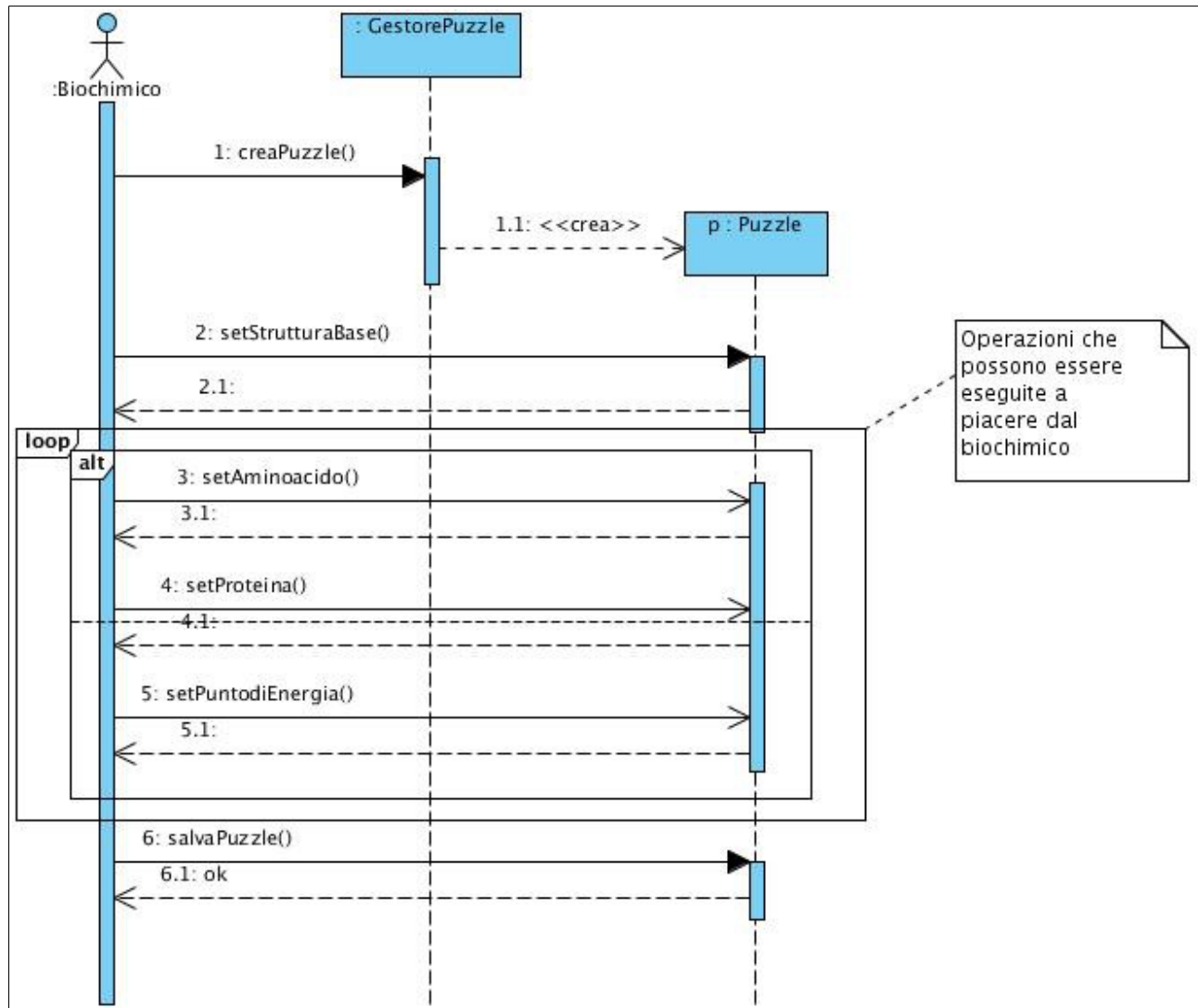


## 5.6 Scelta lingua

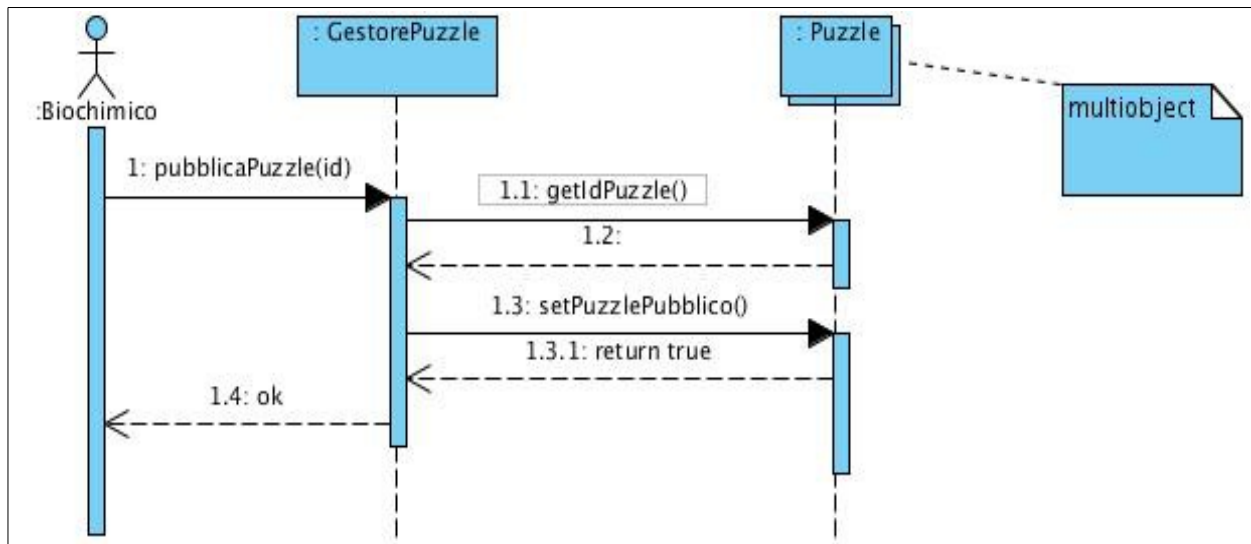




## 5.7 Crea puzzle



## 5.8 Pubblica



## 5.9 Cancella puzzle

