

## Sommario

<b>Introduzione .....</b>	<b>5</b>
Problema e a chi si rivolge il prodotto .....	5
Requisiti .....	5
<b>Glossario .....</b>	<b>6</b>
Definizioni .....	6
<b>Iterazione 1 .....</b>	<b>8</b>
UC1: Creazione della competizione .....	8
Diagramma di Sequenza di Sistema .....	9
Contratti delle Operazioni.....	9
UC2: Tesseramento degli Atleti .....	10
Diagramma di Sequenza di Sistema .....	11
Contratti delle Operazioni.....	11
UC3: Iscrizione dell'Atleta .....	13
Diagramma di Sequenza di Sistema .....	13
Contratti delle Operazioni.....	14
Modello di Dominio .....	14
Progettazione.....	16
Diagrammi di Sequenza UC1.....	16
Diagrammi di Sequenza UC2.....	17
Diagrammi di Sequenza UC3.....	18
Diagramma delle classi.....	19
Testing.....	20
Testing UC1 .....	20
Testing UC2 .....	20
Testing UC3 .....	21
<b>Iterazione 2 .....</b>	<b>22</b>
UC4: Gestione della formula di gara .....	22
Diagramma di Sequenza di Sistema .....	23
Contratti delle Operazioni.....	23
UC5: Accettazione degli Atleti .....	24
Diagramma di Sequenza di Sistema .....	25
Contratti delle Operazioni.....	25
Modello di Dominio .....	26

Progettazione.....	27
Diagrammi di Sequenza UC4.....	27
Diagrammi di Sequenza UC5.....	29
Diagramma delle Classi.....	30
Testing.....	31
Testing UC4.....	31
Testing UC5.....	31
<b>Iterazione 3 .....</b>	<b>32</b>
UC6: Creazione dei gironi.....	32
Diagramma di sequenza.....	33
Contratti delle Operazioni.....	33
UC7: Gestione degli assalti dei gironi.....	34
Diagramma di Sequenza di Sistema.....	35
Contratti delle Operazioni.....	35
Modello di Dominio .....	36
Progettazione.....	37
Diagrammi di Sequenza UC6.....	37
Diagrammi di Sequenza UC7.....	38
Diagramma delle Classi.....	40
Testing.....	41
Testing UC6.....	41
Testing UC7.....	41
<b>Iterazione 4 .....</b>	<b>42</b>
UC7.1: Creazione e visualizzazione classifica dopo i gironi.....	42
Diagramma di Sequenza di Sistema.....	43
Contratti delle Operazioni.....	43
UC8: Creazione del tabellone ad eliminazione diretta .....	44
Diagramma di Sequenza di Sistema.....	45
Contratti delle Operazioni.....	45
UC9: Gestione degli assalti ad eliminazione diretta .....	46
Diagramma di Sequenza di Sistema.....	47
Contratti delle Operazioni.....	47
Modello di Dominio .....	49
Progettazione.....	50
Diagrammi di Sequenza UC7.1.....	50

Diagrammi di Sequenza UC8.....	51
Diagrammi di Sequenza UC9.....	52
Diagramma delle Classi .....	54
Testing.....	55
Testing UC7.1 .....	55
Testing UC8 .....	55
Testing UC9 .....	56
<b>Iterazione 5 .....</b>	<b>57</b>
UC10: Generazione del ranking .....	57
Diagramma di Sequenza del Sistema.....	58
Contratti delle Operazioni.....	58
UC11: Visualizzazione delle classifiche .....	59
Diagramma di Sequenza di Sistema .....	59
Contratti delle Operazioni.....	60
Modello di Dominio .....	61
Progettazione.....	62
Diagrammi di Sequenza UC10.....	62
Diagrammi di Sequenza UC11.....	63
Diagramma delle Classi .....	64
Testing.....	65
Testing UC10 .....	65
Testing UC11 .....	66
<b>Interfaccia grafica .....</b>	<b>67</b>
UC1.....	68
UC2.....	69
UC3.....	70
UC4.....	70
UC5.....	70
UC6.....	71
UC7.....	71
UC7.1.....	71
UC8.....	72
UC9.....	72
UC11.....	73
UC10.....	73



## Introduzione

Prevediamo la realizzazione di un'applicazione di gestione di competizioni schermistiche, chiamata ProFencer. L'obiettivo dell'applicazione è gestire tutte le fasi della gara, dall'iscrizione alla pubblicazione degli esiti finali.

### Problema e a chi si rivolge il prodotto

La presenza di un sistema non automatizzato rende praticamente impossibile la gestione di una gara con un gran numero di iscritti. Le gare a livello nazionale prevedono la partecipazione di circa un migliaio di atleti. Esistono già dei sistemi automatizzati che gestiscono tali competizioni, tuttavia possono essere considerati ormai obsoleti, ad esempio non prevedono la generazione automatica del ranking. Il software è rivolto alle federazioni schermistiche o chiunque voglia gestire delle competizioni.

### Requisiti

La gestione delle gare dovrebbe prevedere:

- la creazione della gara con la sua specifica formula
- tesseramento degli atleti
- l'iscrizione degli atleti
- l'accettazione
- la creazione dei gironi
- la gestione degli assalti dei gironi e delle eliminazioni dirette
- generazione del ranking
- le visualizzazioni delle classifiche

# Glossario

## Definizioni

Termine	Definizione
Scherma	La scherma è uno sport o un'arte marziale che coinvolge il combattimento con armi bianche, come la spada, il fioretto e la sciabola. Questo sport ha origini antiche ed è stato sviluppato nel corso dei secoli in diverse forme e stili. La spada è un'arma non convenzionale.
Spada	È un'arma e prevede che i punti siano segnati colpendo l'avversario con la punta della lama. Il bersaglio valido comprende tutto il corpo. La spada è un'arma convenzionale.
Fioretto	È un'arma che prevede che i punti siano segnati colpendo l'avversario con la punta. Il bersaglio del fioretto è costituito da torace, addome, fianchi, schiena. La spada è un'arma convenzionale.
Sciabola	È un'arma che prevede che i punti siano segnati colpendo l'avversario sia di punta che di taglio. Il bersaglio valido è tutto il corpo dalla cinta in su.
Assalto	Il combattimento viene denominato Assalto e dura 3 minuti nelle fasi iniziali delle competizioni dove la suddivisione degli assalti è definita in gironi all'italiana, mentre dura 9 minuti (3 tempi da 3 minuti) nella successiva fase di scontri diretti.
Stoccata	Viene definito il punto che si può infliggere in un combattimento
Categorie	Nella scherma ci sono molte categorie: Prime lame (età=10 anni) Maschietti/bambine (età=11 anni) Giovanissimi/e (età=12 anni) Ragazzi/e (età= 13 anni) Allievi/e (età=14 anni) Cadetti/e (15<=età<=17) Giovani (18<=età<=20) Assoluti (età >= 15 anni) Master (over 24)
Tirare	Verbo utilizzato per indicare lo svolgimento di tale sport.
Schermidore	Praticante della scherma.
Girone	Struttura di competizione in cui un gruppo di individui si scontrano tra loro in una serie di assalti. I gironi nella scherma sono composti al massimo da 7 atleti (eccetto per i Master dove i

	gironi possono essere composti al più da 8 schermidori)
Amministratore	Personale della Federazione Italiana Scherma cui lavoro è quello di organizzare i calendari e creare le competizioni.
Ufficiale di gara	Figura che gestisce effettivamente il fluire della competizione attraverso il software. Si trova nel luogo di gara e si interfaccia con un computer in cui è in esecuzione ProFencer.
Ranking	Il ranking è un punteggio posseduto da ogni atleta, ci indica il suo valore sportivo. Viene modificato ad ogni gara disputata.

## Iterazione 1

La prima iterazione implementa gli scenari principali di successo dei casi d'uso:

- UC1: Creazione della competizione
- UC2: Tesseramento dell'atleta
- UC3: Iscrizione dell'atleta

Per ogni caso d'uso vengono effettuate l'analisi, la progettazione, sviluppo del codice e testing.

### UC1: Creazione della competizione

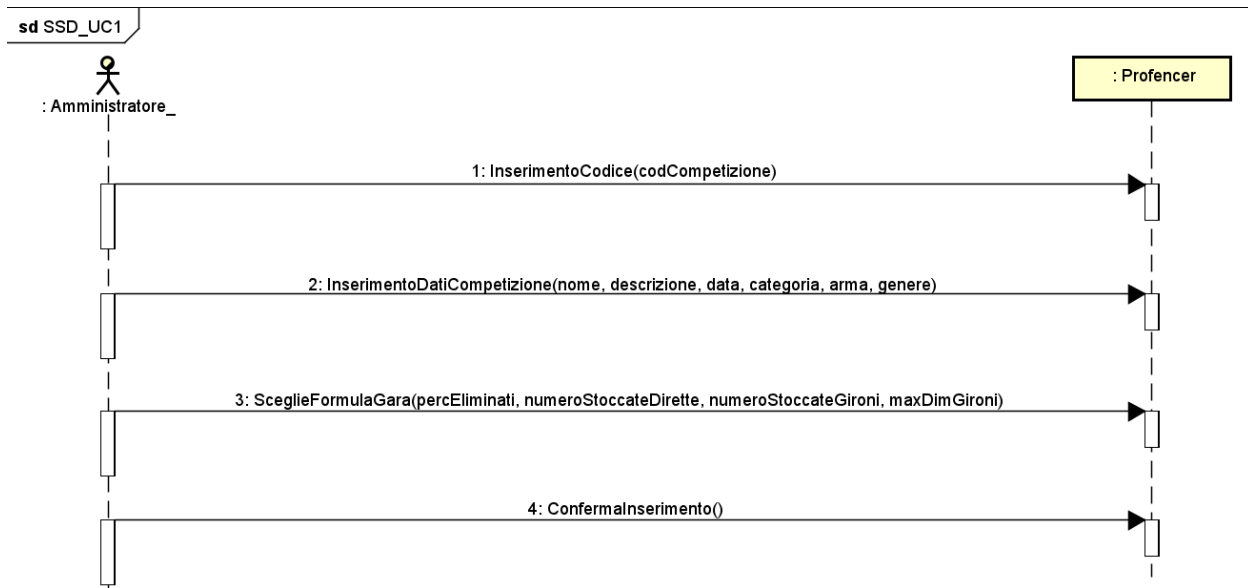
<b>Nome caso d'uso</b>	Creazione della competizione
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Amministratore del sistema
<b>Parti interessate ed interessi</b>	-Amministratore: vuole gestire la creazione della nuova gara con la relativa formula di gara
<b>Pre-condizioni</b>	La gara deve far parte del calendario fatto dalla FIS (Federazione Italiana Scherma).
<b>Garanzia di successo</b>	Le informazioni riguardanti la nuova competizione sono inserite nel sistema.
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"><li>1. L'amministratore vuole creare una nuova competizione</li><li>2. L'amministratore sceglie l'attività "crea competizione"</li><li>3. L'amministratore inserisce il nome/codice della competizione</li><li>4. L'amministratore inserisce diversi dati sulla gara:<ul style="list-style-type: none"><li>• Descrizione</li><li>• Data e ora di inizio</li><li>• Categorie alle quali è rivolta la gara</li></ul></li><li>5. Sceglie la formula di competizione (Il numero di stoccate negli assalti dei gironi, percentuale di atleti che passano alla fase di eliminazione diretta, Il numero di stoccate negli assalti ad eliminazione diretta)</li><li>6. L'amministratore indica di aver finito</li></ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"><li>1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema</li><li>2. Il Sistema ripristina lo stato</li></ol> <p>3a. L'amministratore inserisce nome/codice di una competizione già stata svolta o non presente nel calendario.</p> <ol style="list-style-type: none"><li>1. Il sistema genera un messaggio di errore</li><li>2. L'amministratore ripete il passaggio 3</li></ol>
<b>Requisiti speciali</b>	



<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Legata al numero di gare nel calendario ufficiale
<b>Varie</b>	

### Diagramma di Sequenza di Sistema

Il diagramma di sequenza di sistema illustra il corso degli eventi di input e output tra ProFencer e l'Amministratore:



### Contratti delle Operazioni

Tramite i contratti delle operazioni descriviamo gli eventi scaturiti dalle operazioni di sistema individuate nell'SSD.

#### 1. *Inserimento Codice*

##### **Operazione:**

InserimentoCodice (codCompetizione)

##### **Riferimenti:**

Caso d'uso: Crea nuova competizione

##### **Pre-condizioni:** -

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

- È stata creata un'istanza c di Competizione
- Gli attributi di c sono stati inizializzati
- Viene dato il codice a c
- c diventa corrente

## 2. Inserimento Dati Competizione

**Operazione:**

InserimentoDatiCompetizione(nome, descrizione, data, categoria, arma, genere)

**Riferimenti:**

Caso d'uso: Crea nuova competizione

**Pre-condizioni:**

- È stata già creata una nuova istanza di Competizione
- c è corrente

**Post-condizioni:**

- È in corso l'inserimento dei dati dell'istanza c di Competizione

## 3. Scegli la Formula di Gara

**Operazione:**

ScegliFormulaGara(percEliminati, numeroStoccateDirette, numeroStoccateGironi, maxDimGironi)

**Riferimenti:**

Caso d'uso: Crea nuova competizione

**Pre-condizioni:****Post-condizioni:**

- Si salva l'oggetto formula di gara nella competizione corrente

## 4. ConfermaInserimento

**Operazione:**

ConfermaInserimento()

**Riferimenti:**

Caso d'uso: Crea nuova competizione

**Pre-condizioni:** - Tutti i dati sono stati inseriti

**Post-condizioni:**

- Si salva l'istanza c di Competizione e la si associa a ProFencer tramite la lista "competizioni"

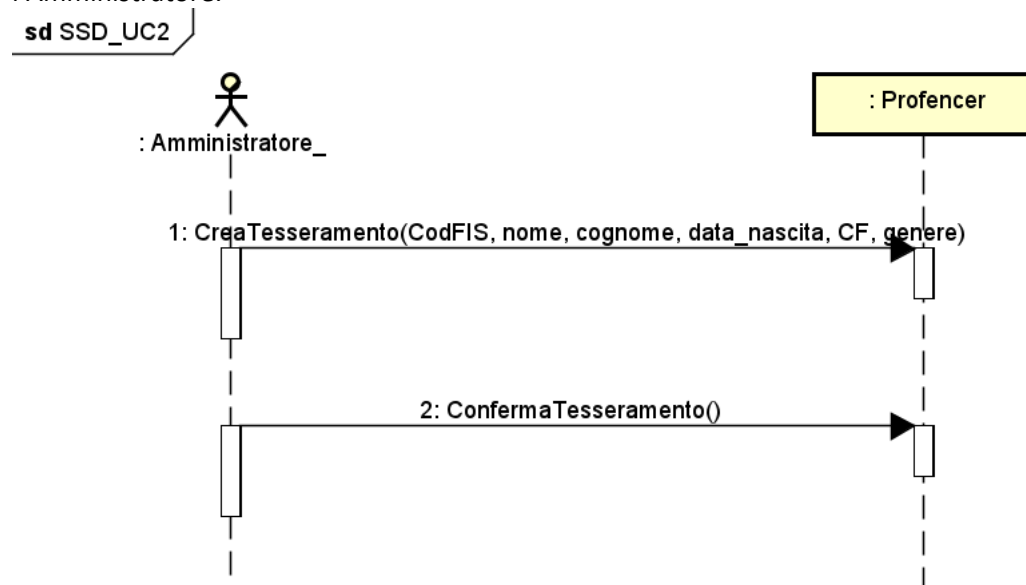
## UC2: Tesseramento degli Atleti

<b>Nome caso d'uso</b>	Tesseramento atleta
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Amministratore del sistema
<b>Parti interessate ed interessi</b>	-Amministratore: vuole tesserare un nuovo atleta -Atleta: vuole tesserarsi alla FIS (Federazione Italiana Scherma)
<b>Pre-condizioni</b>	
<b>Garanzia di successo</b>	Le informazioni riguardanti il nuovo atleta sono inserite nel sistema.
<b>Scenario principale di successo</b>	1. L'atleta va fisicamente da un amministratore con il volere di tesserarsi

	<ol style="list-style-type: none"> <li>2. L'amministratore richiede all'atleta i dati necessari al tesseramento (nome, cognome, data di nascita, CF, certificato medico)</li> <li>3. L'amministratore inserisce nel sistema i dati</li> <li>4. L'amministratore indica di aver completato</li> </ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"> <li>1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema</li> <li>2. Il Sistema ripristina lo stato</li> </ol> <p>3a. Il sistema riceve dati non validi</p> <ol style="list-style-type: none"> <li>1. Il sistema lo notifica all'amministratore</li> <li>2. L'amministratore ripete il passaggio 2</li> </ol> <p>3b. Il sistema vede che esiste già un tesseramento valido per l'atleta che sta provando a tesserarsi</p> <ol style="list-style-type: none"> <li>1. Il sistema lo notifica all'amministratore</li> </ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Legata al numero di atleti che si iscrivono annualmente
<b>Varie</b>	

### Diagramma di Sequenza di Sistema

Il diagramma di sequenza di sistema illustra il corso degli eventi di input e output tra ProFencer e l'Amministratore:



## 1. *CreaTesseramento*

**Operazione:**

CreaTesseramento(CodFIS, nome, cognome, data\_nascita, CF, genere)

**Riferimenti:**

Caso d'uso: Crea Tesseramento

**Pre-condizioni:****Post-condizioni:**

- È stata creata un'istanza a di Atleta
- Sono stati assegnati gli attributi di a
- a diventa corrente

## 2. *ConfermaTesseramento*

**Operazione:**

ConfermaTesseramento()

**Riferimenti:**

Caso d'uso: Crea Tesseramento

**Pre-condizioni:**

- È stata creata una nuova istanza a di Atleta
- a è corrente

**Post-condizioni:**

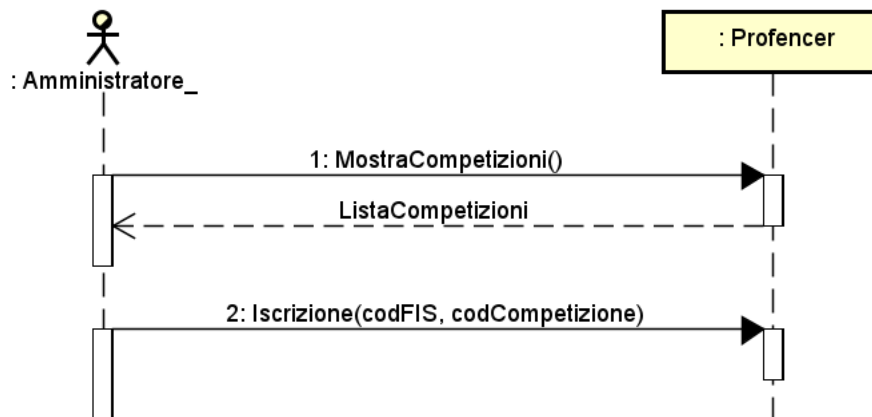
- Si salva l'istanza a di Atleta e la si associa a ProFencer tramite la lista "tesserati"

### UC3: Iscrizione dell'Atleta

<b>Nome caso d'uso</b>	Iscrizione dell'atleta alla competizione
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Amministratore
<b>Parti interessate ed interessi</b>	Atleta: vuole iscriversi alla gara Amministratore: iscrive l'atleta alla gara
<b>Precondizioni</b>	Deve esistere una gara a cui iscriversi. L'atleta deve essere tesserato alla federazione e appartenere ad una categoria.
<b>Garanzia di successo</b>	Le informazioni riguardanti l'iscrizione dell'atleta sono inserite nel sistema.
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"><li>1. L'amministratore accede al sistema</li><li>2. L'amministratore clicca su "iscrizione"</li><li>3. Il sistema mostra tutte le competizioni all'amministratore</li><li>4. L'amministratore sceglie a quale gara iscrivere l'atleta</li><li>5. Il sistema verifica che l'atleta sia tesserato e che la competizione a cui vuole iscriversi sia aperta alla sua categoria</li><li>6. Il sistema notifica all'atleta il successo dell'operazione</li></ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"><li>1. L'amministratore riavvia il software e ripristina lo stato precedente del Sistema</li><li>2. Il Sistema ripristina lo stato</li></ol> <p>5a. L'amministratore sta provando ad iscrivere un atleta ad una competizione non consentita</p> <ol style="list-style-type: none"><li>1. Il sistema genera un messaggio di errore</li><li>2. Si ritorna allo stato 3</li></ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Legata al numero medio di gare a cui l'atleta si iscrive
<b>Varie</b>	

### Diagramma di Sequenza di Sistema

Il diagramma di sequenza di sistema illustra il corso degli eventi di input e output tra ProFencer e l'Atleta:



## Contratti delle Operazioni

### 1. *MostraCompetizioni*

**Operazione:**

MostraCompetizioni()

**Riferimenti:**

Caso d'uso: Iscrizione dell'atleta

**Pre-condizioni:**

- Sono presenti istanze di Competizione

**Post-condizioni:** -

### 2. *Iscrizione*

**Operazione:**

Iscrizione(codFIS, codCompetizione)

**Riferimenti:**

Caso d'uso: Iscrizione dell'atleta

**Pre-condizioni:**

- L'atleta deve risultare tesserato
- La competizione deve esistere

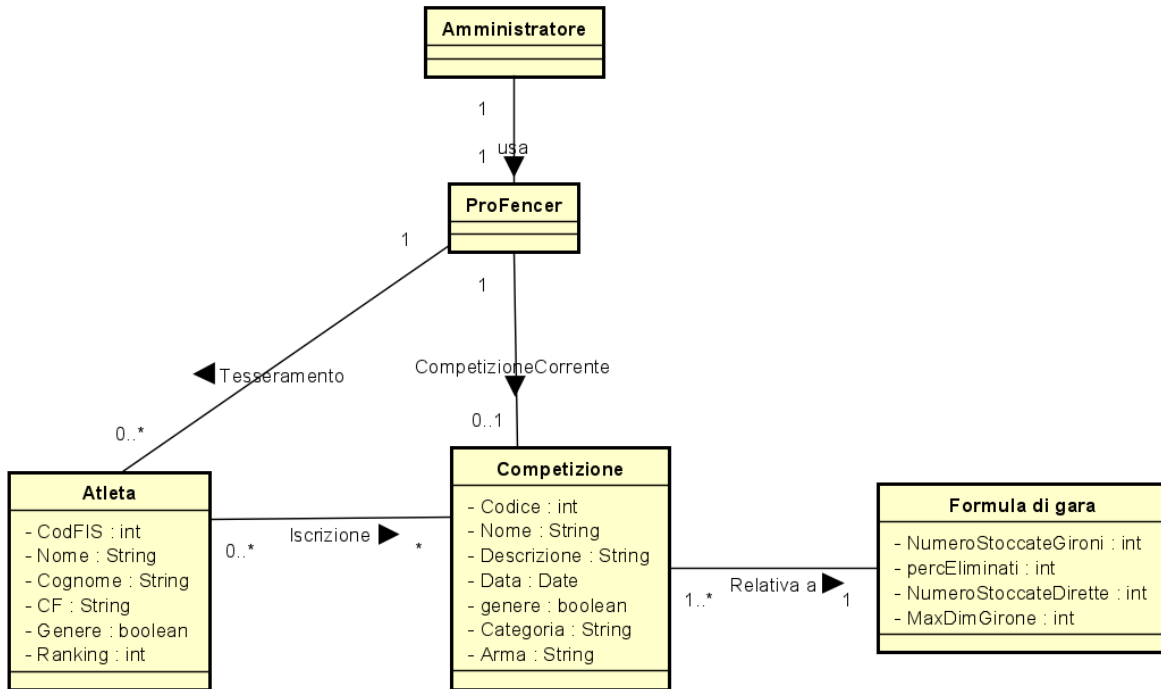
**Post-condizioni:**

- L'atleta viene aggiunto alla lista di competizione "iscrizioni"

## Modello di Dominio

Le classi concettuali identificate nella prima iterazione sono:

- Amministratore: attore primario, che interagisce con il software
- ProFencer: il software vero e proprio
- Atleta: viene tesserato dal software su richiesta dell'amministratore
- Competizione: oggetto posseduto dal software, a cui si iscrivono gli atleti
- Formula di gara: oggetto posseduto dal software, a cui si riferisce la competizione

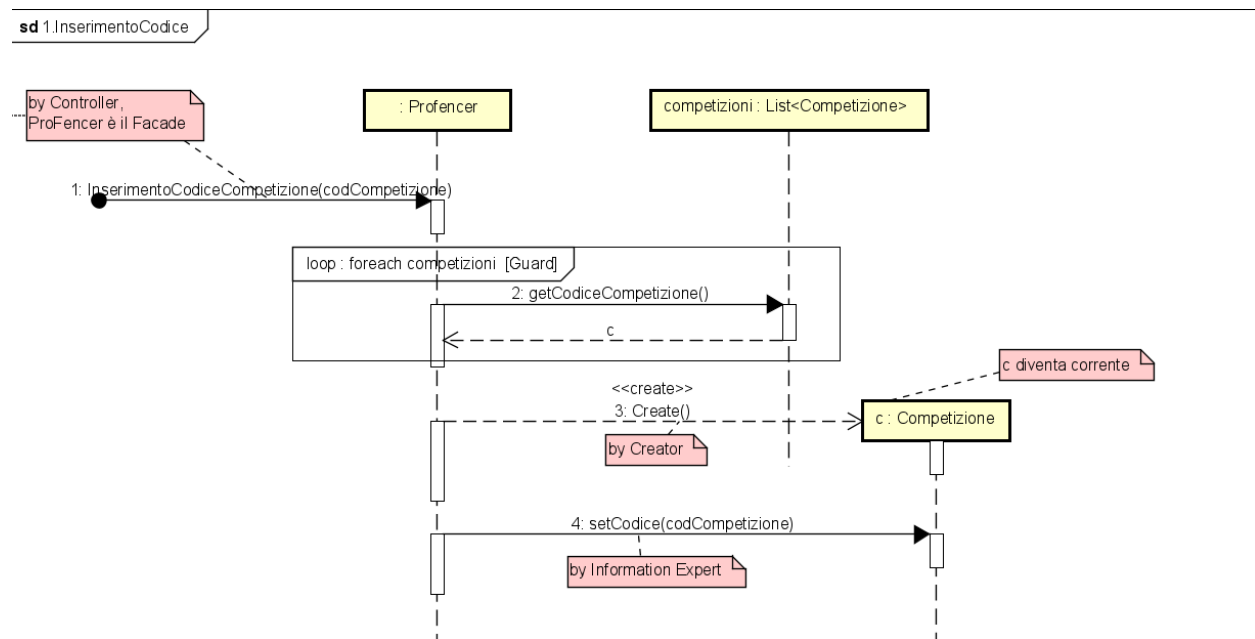


## Progettazione

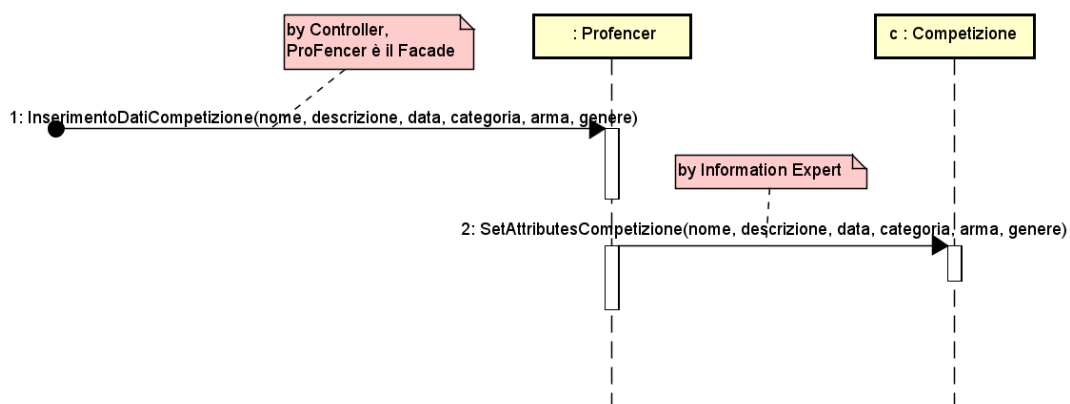
### Diagrammi di Sequenza UC1

ProFencer tiene traccia di tutte le competizioni; dunque, gli è stata affidata la responsabilità di creare nuovi oggetti di tipo Competizione (per il pattern Creator) e di inserirne gli attributi per ogni istanza. Gli oggetti di tipo Competizione contengono oggetti di tipo FormuladiGara; dunque, la creazione di oggetti di tipo FormuladiGara è delegata alla classe Competizione. È possibile scegliere tra una formula di gara già esistente o crearne una nuova.

#### InserimentoCodice

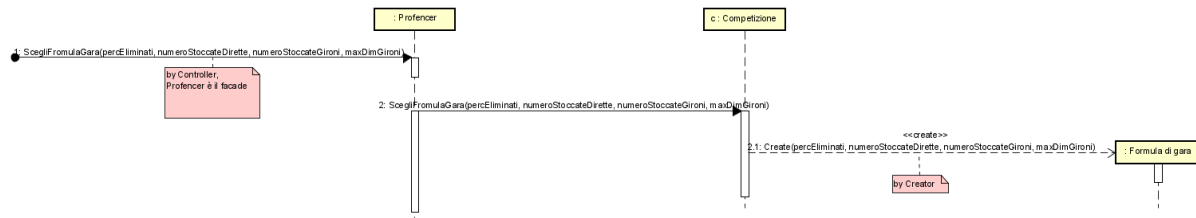


#### InserimentoDatiCompetizione



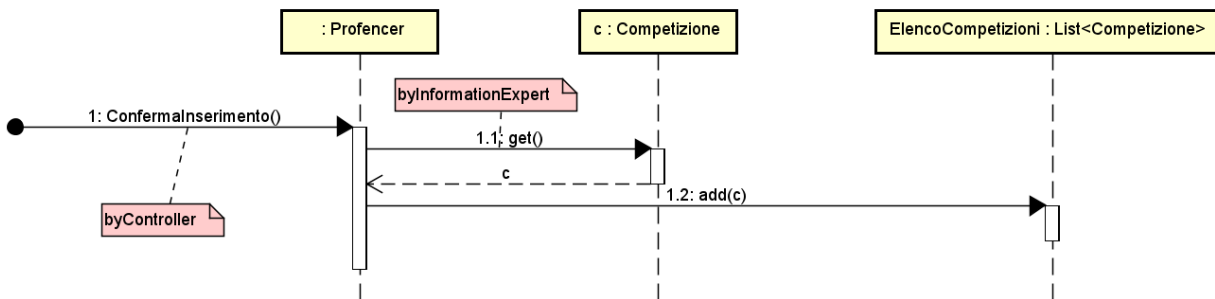


### ScegliFormulaGara



### ConfermaInserimento

#### sd 4. ConfermaInserimento

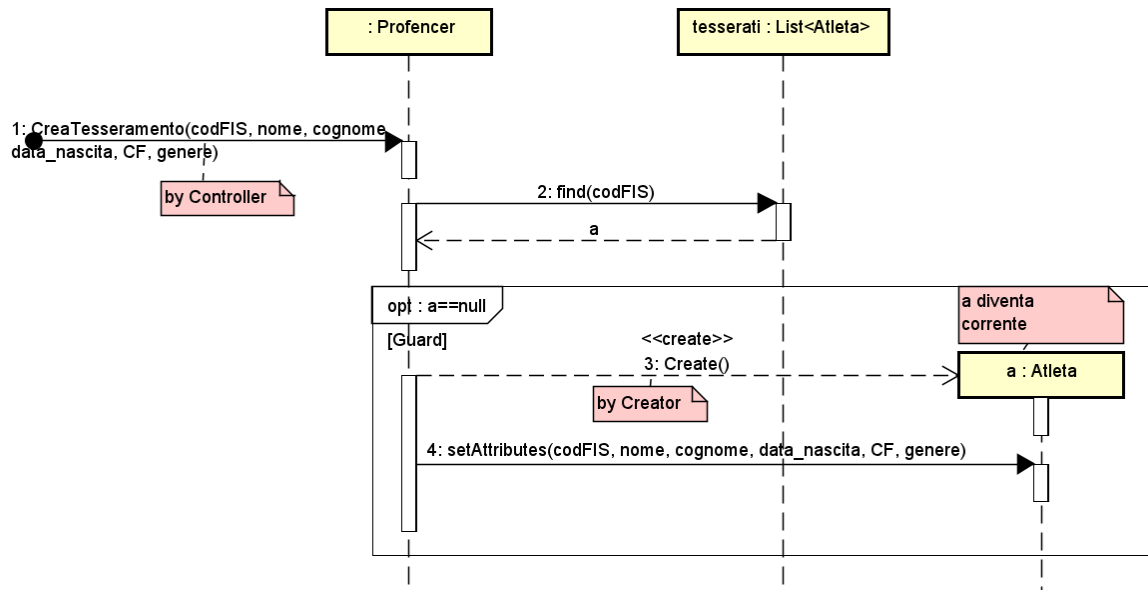


L'oggetto *c* di tipo *Competizione* si riferisce alla *competizioneCorrente* del sistema.

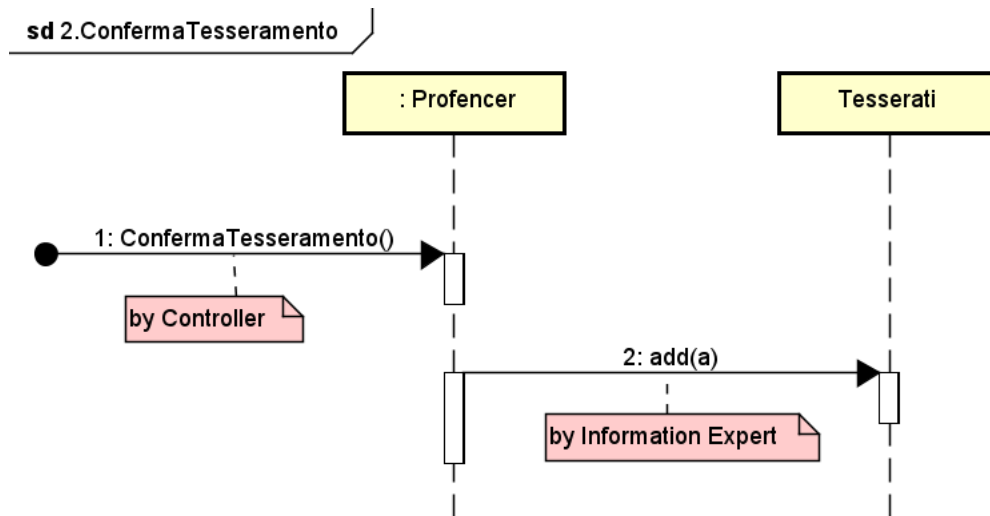
## Diagrammi di Sequenza UC2

### CreaTesseramento

#### sd 1. CreaTesseramento



### ConfermaTesseramento

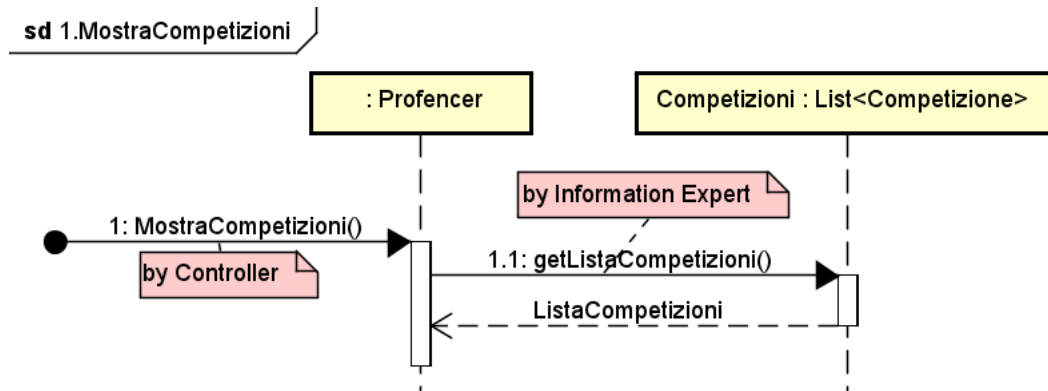


ProFencer tiene traccia di tutte gli atleti; dunque, gli è stata affidata la responsabilità di creare nuovi oggetti di tipo *Atleta* (per il pattern *Creator*) e di inserirne gli attributi per ogni istanza. *Tesserati* è di tipo *List<Atleta>*.

L'oggetto *a* di tipo *Atleta* si riferisce all'*atletaCorrente* del sistema.

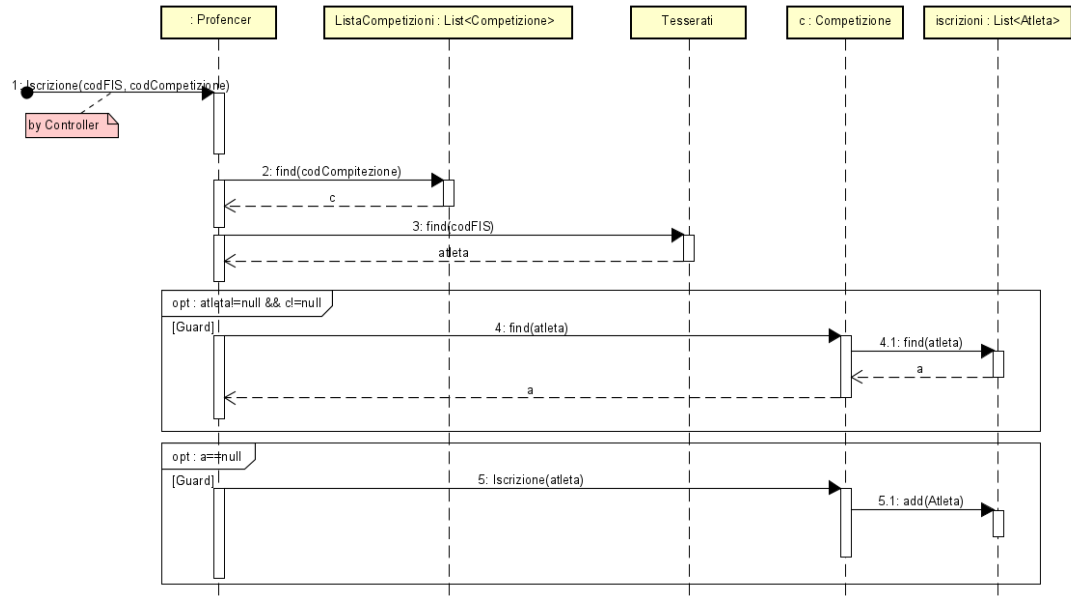
### Diagrammi di Sequenza UC3

### MostraCompetizioni



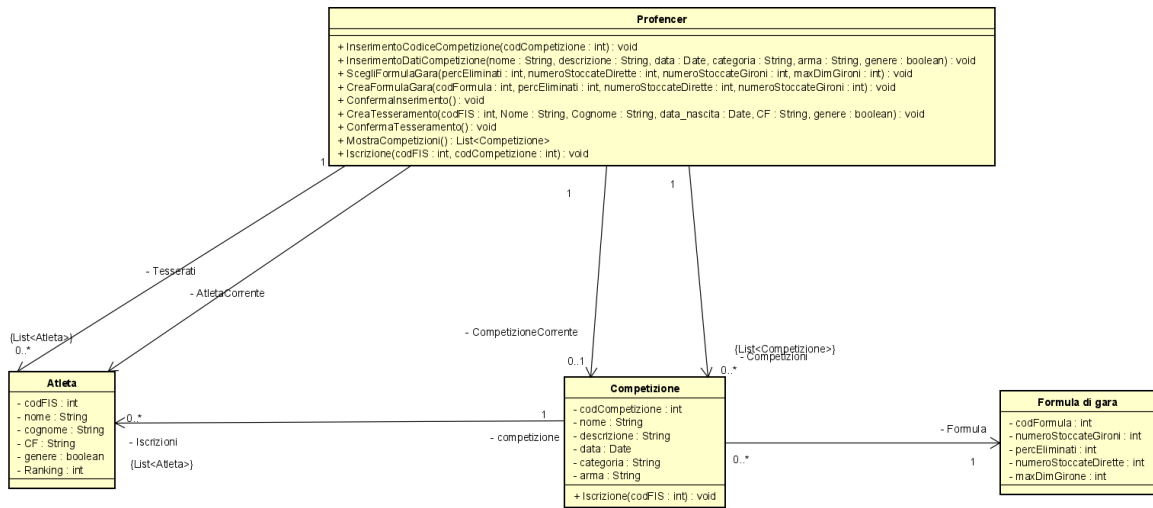
## Iscrizione

sd2.Iscrizione



## Diagramma delle classi

pkgModello di Progetto



## Testing

### Testing UC1

```
@test
public void UC1(){
    proFencer.caricaDati();
    try {
        proFencer.InserimentoCodiceCompetizione(codCompetizione:10);
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
    try {
        proFencer.InserimentoDatiCompetizione(nome:"gara1", descrizione:"descrizione1", data:null, categoria:"U14", arma:"Spada", genere:true);
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
    try {
        proFencer.ScegliFormulaGara(percEliminati:0, numeroStoccateDirette:15, numeroS_5, 7);
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }

    try {
        proFencer.ConfermaInserimento();
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
        // TODO: handle exception
    }
    //System.out.println(proFencer.getCompetizioneCorrente().toString());
    System.out.println(proFencer.getCompetizioni());
}
//Il test funziona correttamente. Viene sollevata un'eccezione se si prova ad inserire una competizione
//con codice già presente nel sistema e se si inseriscono dati della formula di gara non possibili
```

### Testing UC2

```
@Test
public void UC2(){
    proFencer.caricaDati();
    try {
        proFencer.CreaTesseramento(codFIS:1, Nome:"mario", Cognome:"rossi", data_nascita:null, CF:"aaaaaaa", genere:true);
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
    try {
        proFencer.ConfermaTesseramento();
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
    System.out.println(proFencer.getAtletaCorrente().toString());
    System.out.println(proFencer.getTesserati());
}
//Il test funziona correttamente. Viene sollevata un'eccezione se si prova a tesserare un atleta già tesserato
```

## Testing UC3

```
@Test
public void testIscrizione() {
    proFencer.caricaDati();
    try {
        //dobbiamo tesserare un atleta e creare una competizione
        proFencer.CreaTesseramento(codFIS:1, Nome:"mario", Cognome:"rossi", data_nascita:null, CF:"aaaaaaa", genere:true);
        proFencer.ConfermaTesseramento();
        proFencer.Iscrizione(codFIS:1, codCompetizione:1);
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}

//Il test funziona correttamente. Viene sollevata un'eccezione se si prova a iscrivere un atleta non tesserato
// se si iscrive un atleta già iscritto o se si prova a iscrivere a un c non esistente
```

## Iterazione 2

La prima iterazione implementa gli scenari principali di successo dei casi d'uso:

- UC4: Gestione della formula di gara
- UC5: Accettazione degli atleti

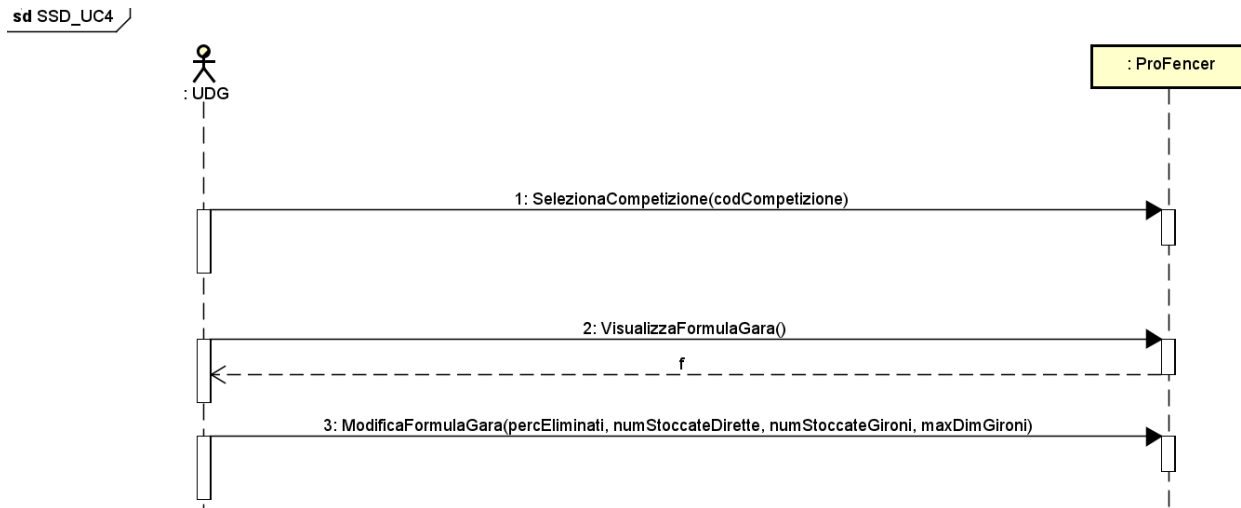
Per ogni caso d'uso vengono effettuate l'analisi, la progettazione, sviluppo del codice e testing.

### UC4: Gestione della formula di gara

<b>Nome caso d'uso</b>	Gestione della formula di gara
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Ufficiale di gara
<b>Parti interessate ed interessi</b>	-Ufficiale di gara: vuole cambiare la formula di gara (quasi sempre l'Ufficiale di gara deve modificare la formula di gara definita precedentemente dall'amministratore per esigenze presentate nel luogo di gara)
<b>Pre-condizioni</b>	La competizione è già stata creata, gli atleti si sono iscritti e si sta effettivamente svolgendo.
<b>Garanzia di successo</b>	Le modifiche alla formula di gara sono state effettuate correttamente.
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"><li>1. L'Ufficiale di gara seleziona la competizione che sta gestendo in quel momento.</li><li>2. Viene visualizzata la formula di gara definita dall'amministratore al momento della creazione della competizione.</li><li>3. L'Ufficiale di gara sceglie cosa modificare.</li><li>4. Conferma delle modifiche eseguite.</li></ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"><li>1. L'UDG riavvia il software e ripristina lo stato precedente del Sistema</li><li>2. Il Sistema ripristina lo stato</li></ol> <p>3a. UFD inserisce dati non validi</p> <ol style="list-style-type: none"><li>1. Il sistema genera un messaggio di errore</li><li>2. L'amministratore ripete il passaggio 3</li></ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Solitamente all'inizio di ogni competizione.
<b>Varie</b>	

## Diagramma di Sequenza di Sistema

Il diagramma di sequenza di sistema illustra il corso degli eventi di input e output tra ProFencer e l'Ufficiale di Gara:



## Contratti delle Operazioni

### 1. *SelezionaCompetizione*

**Operazione:**

SelezionaCompetizione(codCompetizione)

**Riferimenti:**

Caso d'uso: Gestione della Formula di Gara

**Pre-condizioni:** Deve esistere la competizione.

**Post-condizioni:**

- L'istanza della Competizione *c* diventa corrente

### 2. *VisualizzaFormulaGara*

**Operazione:**

VisualizzaFormulaGara()

**Riferimenti:**

Caso d'uso: Gestione della Formula di Gara

**Pre-condizioni:** *c* deve essere corrente.

**Post-condizioni:**

- *f* viene visualizzata

### 3. *ModificaFormulaGara*

**Operazione:**

ModificaFormulaGara(percEliminati, numStoccateDirette, numStoccateGironi, maxDimGironi)

**Riferimenti:**

Caso d'uso: Gestione della Formula di Gara

**Pre-condizioni:**

**Post-condizioni:**

- Le variabili di f vengono modificate

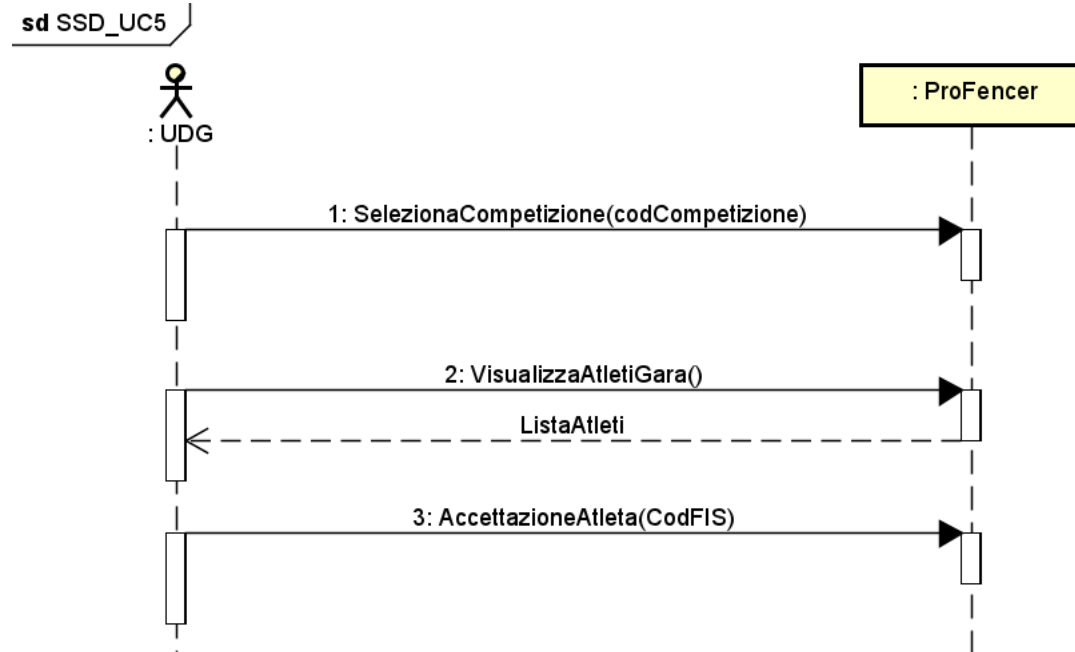
## UC5: Accettazione degli Atleti

<b>Nome caso d'uso</b>	Accettazione degli atleti
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Ufficiale di gara
<b>Parti interessate ed interessi</b>	-Ufficiale di gara: deve confermare la presenza degli atleti nel luogo di gara e verificare che essi siano in grado di competere.
<b>Pre-condizioni</b>	La competizione è già stata creata, gli atleti si sono iscritti e si sta effettivamente svolgendo.
<b>Garanzia di successo</b>	La presenza degli atleti è stata registrata nel sistema.
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"><li>1. L'Ufficiale di gara seleziona la competizione che sta gestendo in quel momento.</li><li>2. Vengono visualizzati tutti gli atleti iscritti alla gara in questione.</li><li>3. UDG, successivamente alla verifica della presenza e dell'idoneità degli atleti, accetta gli atleti alla competizione.</li></ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"><li>1. L'UDG riavvia il software e ripristina lo stato precedente del Sistema</li><li>2. Il Sistema ripristina lo stato</li></ol> <p>3.a. L'atleta è stato precedentemente accettato</p> <ol style="list-style-type: none"><li>1. Il sistema segnala l'errore</li><li>2. Torna allo stato 2</li></ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Solitamente all'inizio di ogni competizione.
<b>Varie</b>	



## Diagramma di Sequenza di Sistema

Il diagramma di sequenza di sistema illustra il corso degli eventi di input e output tra ProFencer e l'Ufficiale di Gara:



## Contratti delle Operazioni

### 1. *selezionaCompetizione*

**Operazione:**

SelezionaCompetizione(codCompetizione)

**Riferimenti:**

Caso d'uso: Accettazione degli atleti

**Pre-condizioni:** Deve esistere la competizione.

**Post-condizioni:**

- L'istanza c della competizione diventa corrente

### 2. *VisualizzaAtleti*

**Operazione:**

VisualizzaAtleti()

**Riferimenti:**

Caso d'uso: Accettazione degli atleti

**Pre-condizioni:** Istanza c di competizione è corrente

**Post-condizioni:**

- Viene visualizzata la lista degli atleti iscritti alla competizione c

### 3. *AccettazioneAtleta*

**Operazione:**

AccettazioneAtleta(codFIS)

**Riferimenti:**

Caso d'uso: Accettazione degli atleti

**Pre-condizioni:**

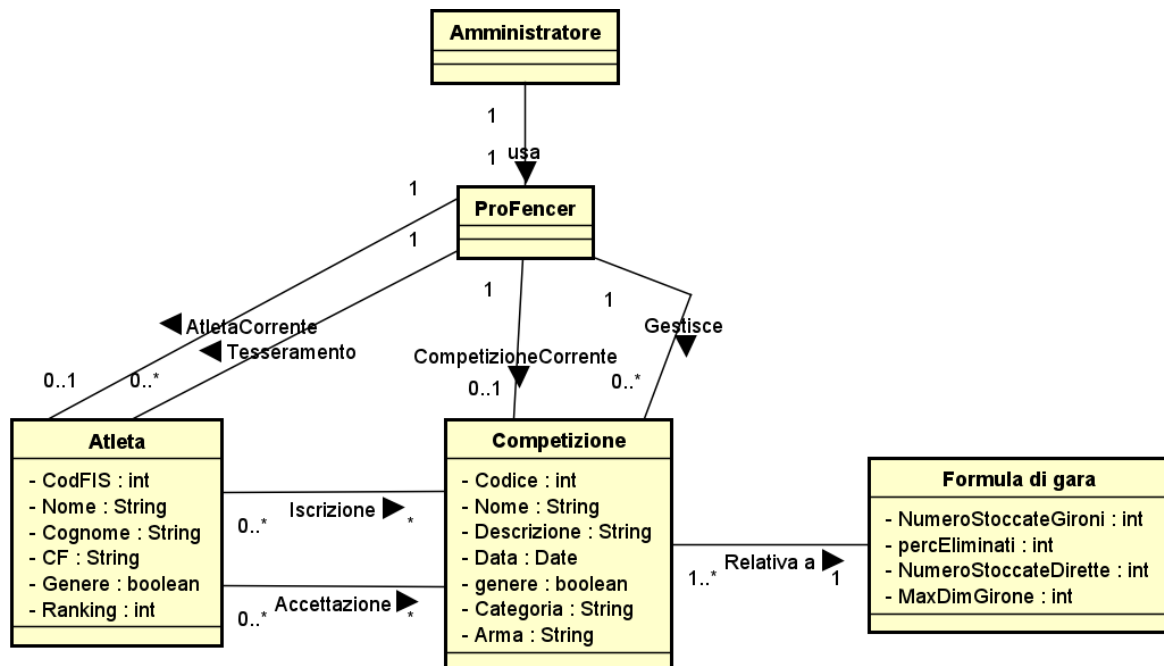
**Post-condizioni:**

- Viene riempita la lista degli atleti accettati "accettazioni"(presenti e idonei)

## Modello di Dominio

Le classi concettuali identificate nella prima iterazione sono:

- Amministratore: attore primario, che interagisce con il software
- ProFencer: il software vero e proprio
- Atleta: viene tesserato dal software su richiesta dell'amministratore
- Competizione: oggetto posseduto dal software, a cui si iscrivono gli atleti
- Formula di gara: oggetto posseduto dal software, a cui si riferisce la competizione

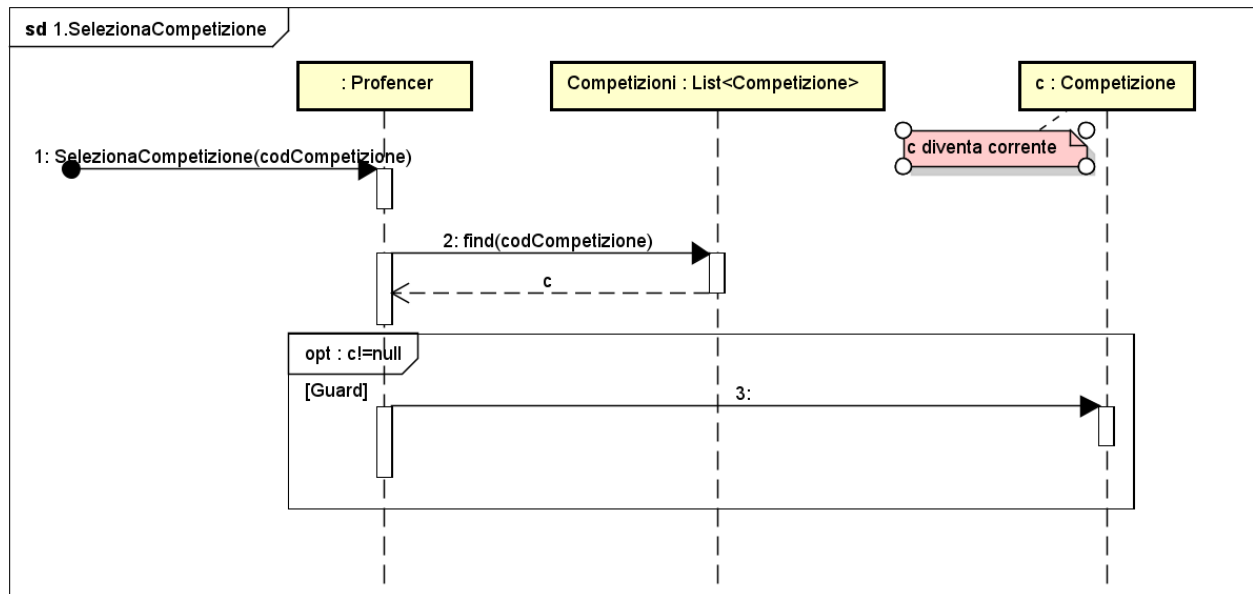


## Progettazione

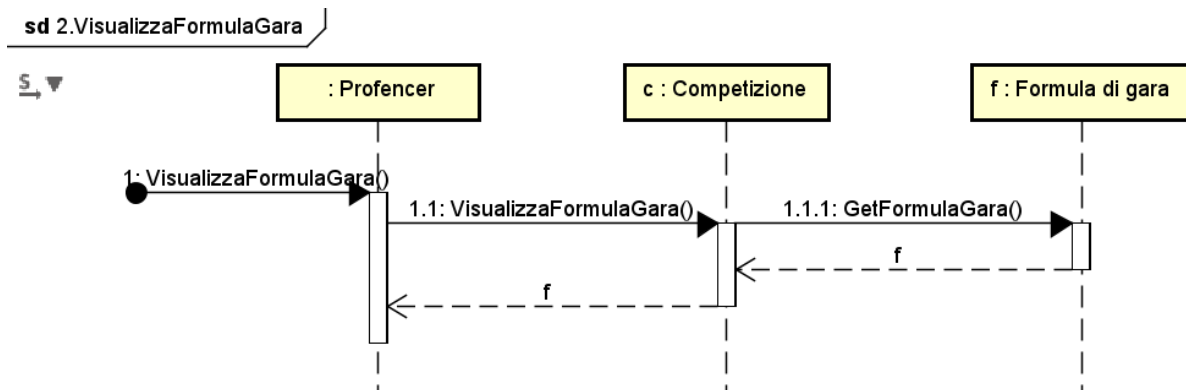
### Diagrammi di Sequenza UC4

ProFencer tiene traccia di tutte le competizioni, dunque cerca un oggetto di tipo *Competizione* in *List<Competizione>*; la competizione trovata diventa la *competizioneCorrente* *c*. Visualizziamo l'oggetto *FormuladiGara* contenuto in *competizioneCorrente* e infine gli oggetti di tipo *Competizione* contengono oggetti di tipo *FormuladiGara*; dunque, la modifica di oggetti di tipo *FormuladiGara* è delegata alla classe *Competizione*.

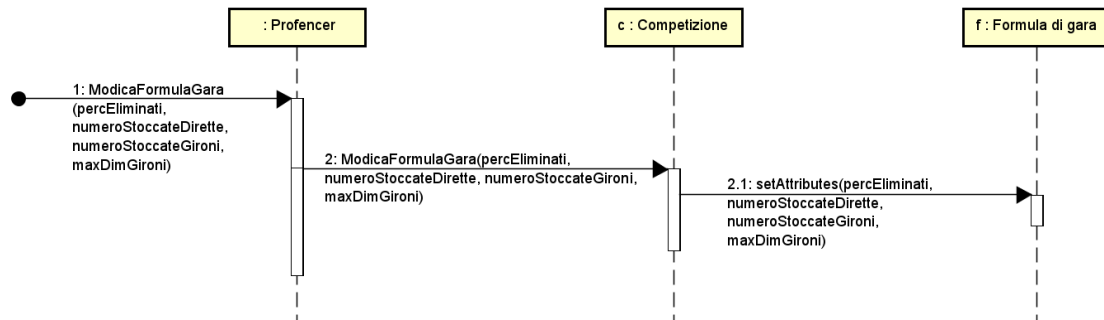
#### SelezioneCompetizione



#### VisualizzaFormulaGara

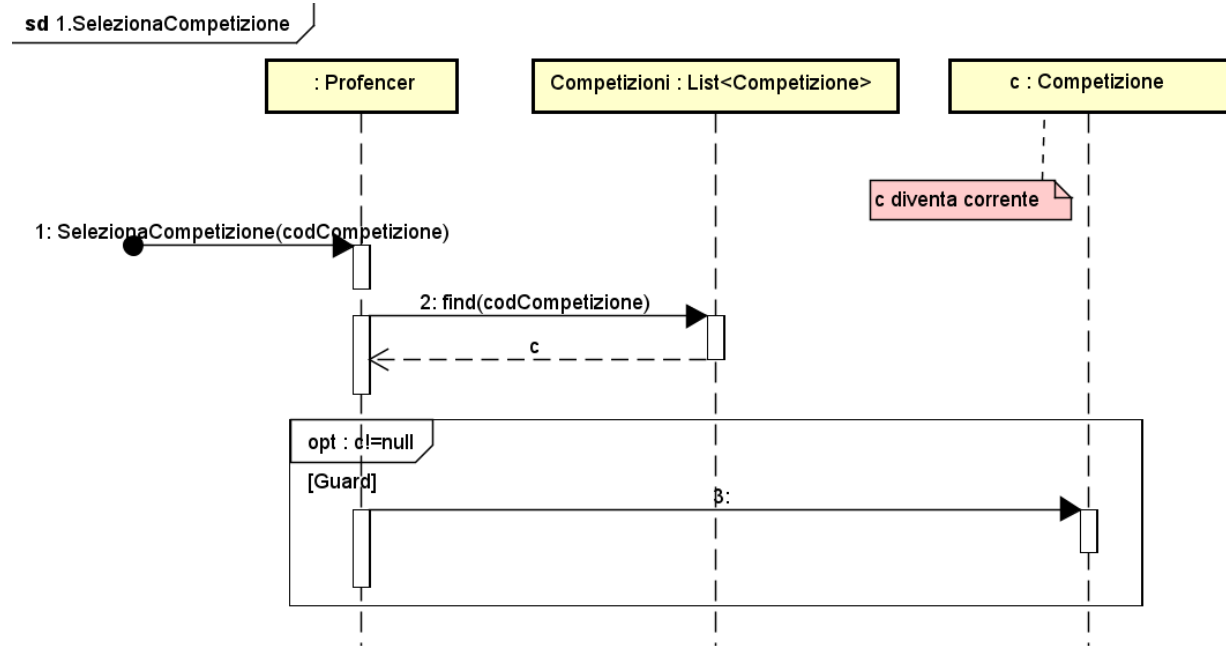


## ModificaFormulaGara

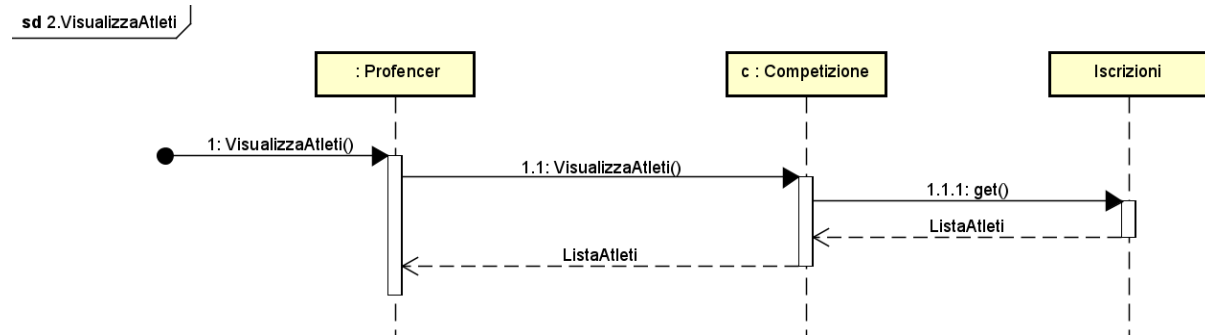


## Diagrammi di Sequenza UC5

### SelezionaCompetizione

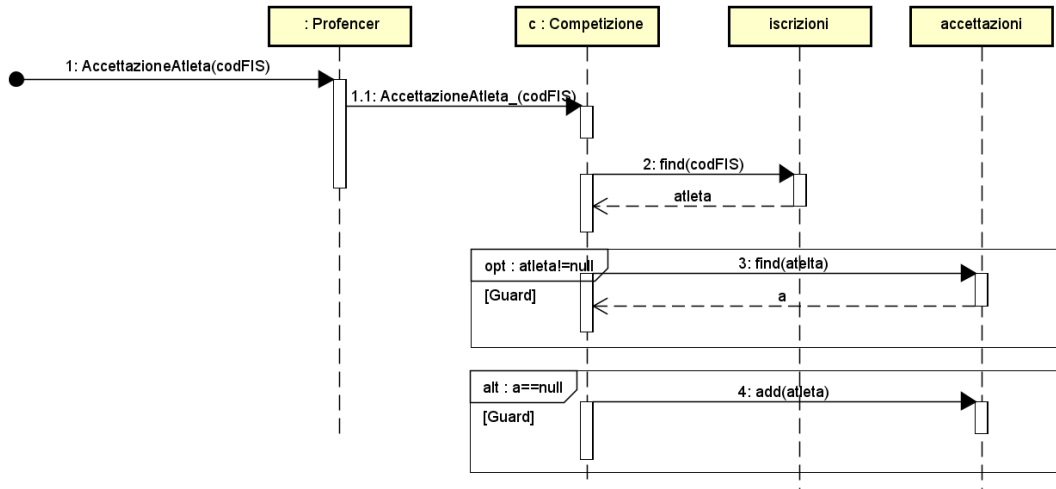


### VisualizzaAtleti



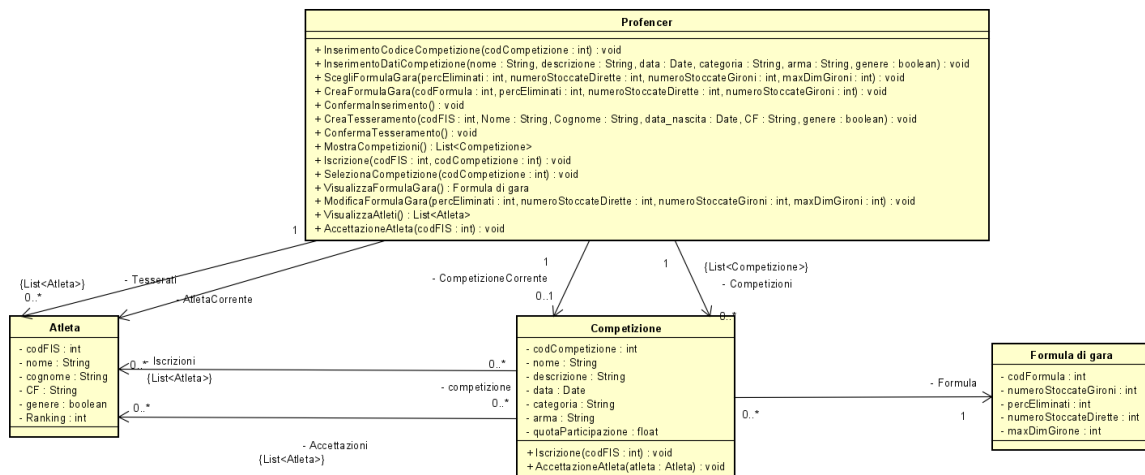
## AccettazioneAtleta

### sd 3.AccettazioneAtleta



Dove *iscrizioni* e *accettazioni* sono di tipo  $\{\text{Map}\langle \text{CodFIS}, \text{Codice} \rangle\}$  tra Competizione e Atleta.

## Diagramma delle Classi



## Testing

### Testing UC4

```
@Test
public void UC4(){
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        System.out.println(proFencer.VisualizzaFormulaGara());
        proFencer.ModificaFormulaGara(percEliminati:0, numeroStoccateDirette:9, numeroS...111, 5);
        System.out.println(proFencer.VisualizzaFormulaGara());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}

//Il test funziona correttamente. Viene sollevata un'eccezione se si inseriscono dati della formula di gara non possibili
```

### Testing UC5

```
@Test
public void UC5(){
    proFencer.caricaDati();
    try {
        proFencer.CreaTesseramento(codFIS:1, Nome:"mario", Cognome:"rossi", data_nascita:null, CF:"aaaaaaa", genere:true);
        proFencer.ConfermaTesseramento();
        proFencer.Iscrizione(codFIS:1, codCompetizione:1);
        proFencer.SelezionaCompetizione(codCompetizione:1);
        System.out.println( proFencer.VisualizzaAtleti());
        proFencer.AccettazioneAtleta(codFIS:1);
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}

//Il test funziona correttamente. Viene sollevata un'eccezione se si prova ad accettare un atleta già accettato o un atleta non iscritto
```

## Iterazione 3

La terza iterazione implementa gli scenari principali di successo dei casi d'uso:

- UC6: Creazione dei gironi
- UC7: Gestione degli assalti dei gironi

Per ogni caso d'uso vengono effettuate l'analisi, la progettazione, sviluppo del codice e testing.

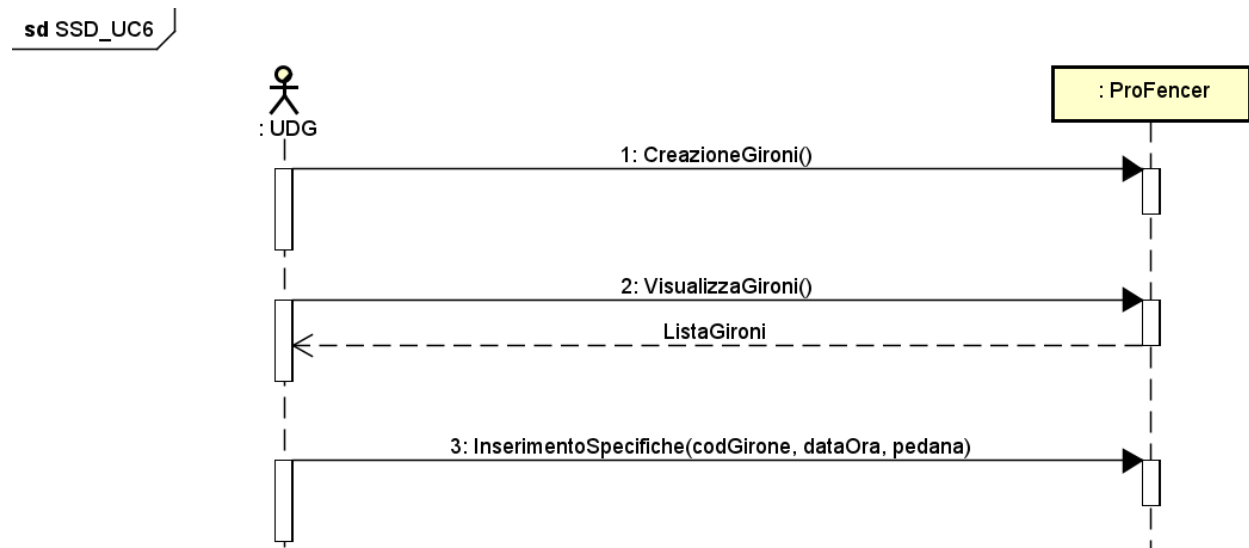
### UC6: Creazione dei gironi

<b>Nome caso d'uso</b>	<b>Creazione dei gironi</b>
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Ufficiale di gara
<b>Parti interessate ed interessi</b>	-Ufficiale di gara: vuole creare i gironi dividendo gli atleti in base alle regole definite
<b>Pre-condizioni</b>	
<b>Garanzia di successo</b>	Le informazioni riguardanti il nuovo girone sono inserite nel sistema.
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"><li>1. L'ufficiale di gara vuole creare i gironi</li><li>2. L'ufficiale di gara sceglie l'attività "creazione gironi"</li><li>3. Profencer restituisce e visualizza la divisione degli atleti per gironi e la lista degli assalti</li><li>4. L'ufficiale di gara inserisce per ogni girone la data, l'orario e pedana.</li><li>5. L'ufficiale indica di aver finito</li></ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"><li>1. L'ufficiale riavvia il software e ripristina lo stato precedente del Sistema</li><li>2. Il Sistema ripristina lo stato</li></ol> <p>3a. L'ufficiale di gara inserisce data/ora/pedana non validi</p> <ol style="list-style-type: none"><li>1. Il sistema genera un messaggio di errore</li><li>2. L'amministratore ripete il passaggio 3</li></ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Legata al numero di gare nel calendario ufficiale
<b>Varie</b>	



## Diagramma di sequenza

Il diagramma di sequenza di sistema illustra il corso degli eventi di input e output tra ProFencer e l'Ufficiale di Gara:



## Contratti delle Operazioni

### 1. CreazioneGironi

**Operazione:**

CreazioneGironi()

**Riferimenti:**

Caso d'uso: Creazione dei gironi

**Pre-condizioni:**

- Deve esserci una competizione corrente

**Post-condizioni:**

- Saranno stati creati tutti i gironi della competizione corrente, i gironi conterranno le informazioni sugli atleti presenti in quel girone e gli assalti. I gironi non avranno ancora i dettagli su data, ora, luogo.

### 2. VisualizzaGironi

**Operazione:**

VisualizzaGironi()

**Riferimenti:**

Caso d'uso: creazione dei gironi

**Pre-condizioni:** i gironi sono stati creati.

**Post-condizioni:**

- I gironi vengono visualizzati senza ancora i loro dettagli

### 3. InserimentoSpecifiche

**Operazione:**

InserimentoSpecifiche(codGirone, dataOra, pedana)

**Riferimenti:**

Caso d'uso: creazione dei gironi

**Pre-condizioni:** i gironi sono stati creati.

**Post-condizioni:**

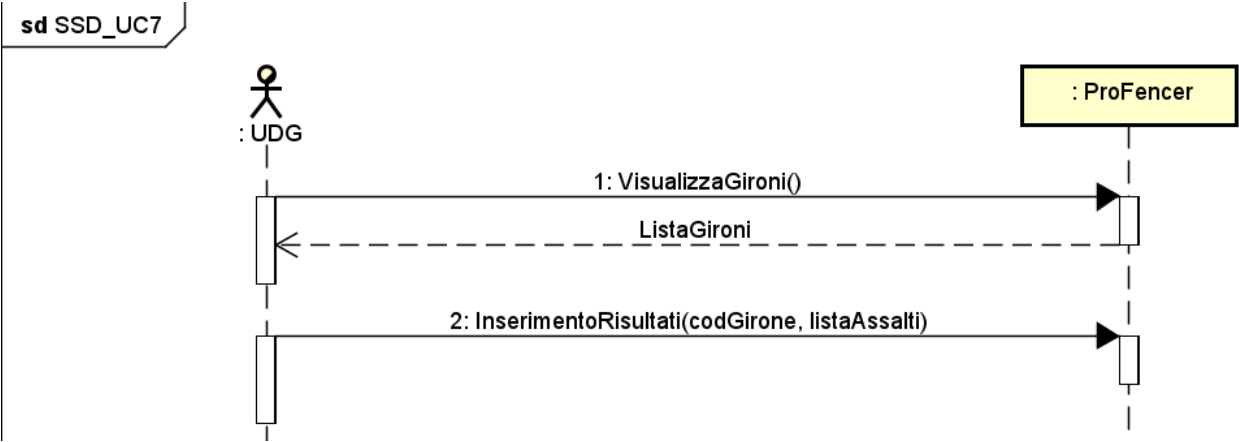
- I gironi sono stati dettagliati.

#### UC7: Gestione degli assalti dei gironi

Nome caso d'uso	Gestione degli assalti dei gironi
Portata	Applicazione ProFencer
Livello	Obiettivo utente
Attore primario	Ufficiale di gara
Parti interessate ed interessi	-L'ufficiale di gara vuole gestire gli assalti all'interno del girone salvandone i risultati
Pre-condizioni	I gironi sono stati creati correttamente.
Garanzia di successo	I risultati verranno salvati correttamente
Scenario principale di successo	<ol style="list-style-type: none"><li>1. L'ufficiale di gara vuole gestire gli assalti dei gironi</li><li>2. L'ufficiale di gara seleziona l'attività gestione assalti</li><li>3. L'ufficiale di gara sceglie il girone da gestire</li><li>4. L'ufficiale di gara inserisce i risultati del girone (popola gli assalti)</li><li>5. ProFencer per ogni atleta calcola le stoccate date, le stoccate ricevute, la differenza tra date e ricevute, il numero di assalti vinti e la percentuale di assalti vinti. Le mostra all'ufficiale di gara.</li></ol>
Estensioni	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"><li>1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema</li><li>2. Il Sistema ripristina lo stato</li></ol> <p>4a. L'ufficiale di gara inserisce dei risultati per il girone non validi.</p> <ol style="list-style-type: none"><li>1. Il sistema genera un messaggio di errore</li><li>2. L'amministratore ripete il passaggio 4</li></ol>
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	

<b>Frequenza di ripetizioni</b>	Legata al numero di gironi della competizione
<b>Varie</b>	

Diagramma di Sequenza di Sistema



Contratti delle Operazioni

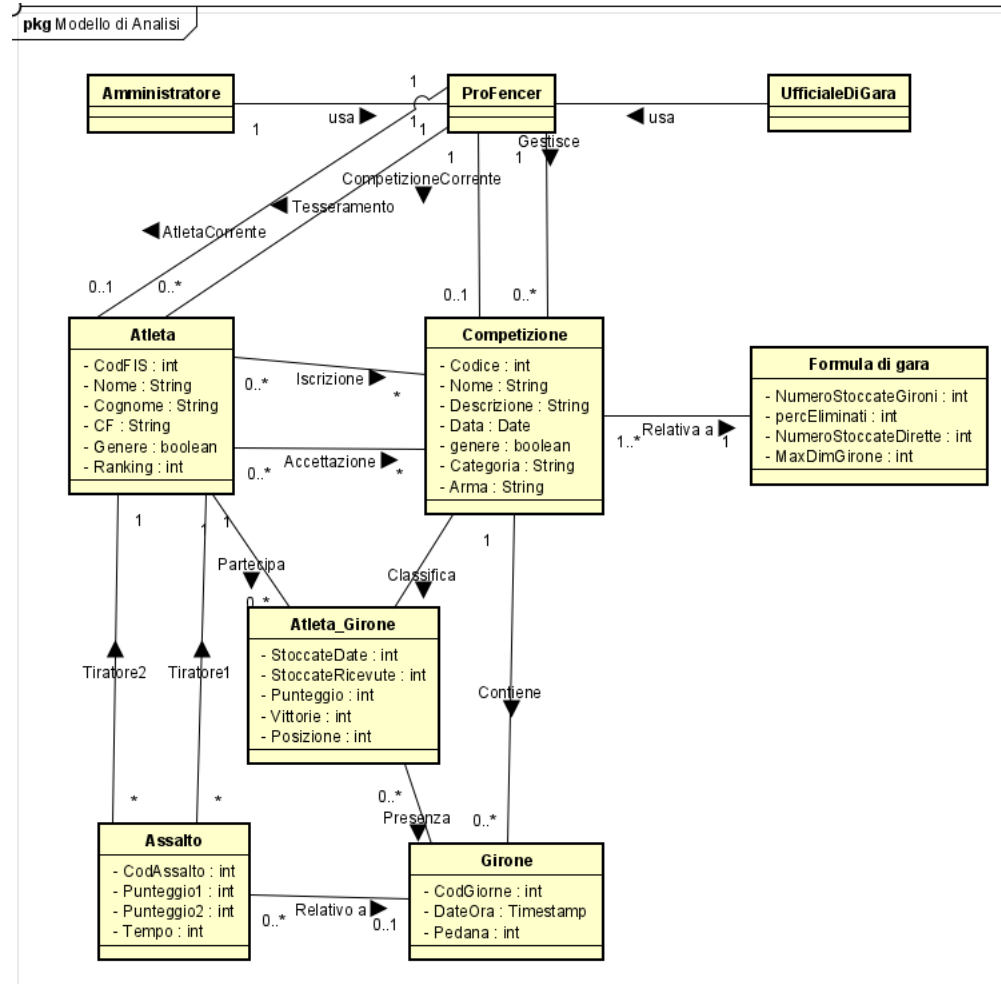
1. VisualizzaGironi

<b>Operazione:</b> VisualizzaGironi()
<b>Riferimenti:</b> Caso d’uso: Gestione degli assalti dei gironi
<b>Pre-condizioni:</b> Devono esistere i gironi.
<b>Post-condizioni:</b> Vengono visualizzati tutti i gironi

2. InserimentoRisultati

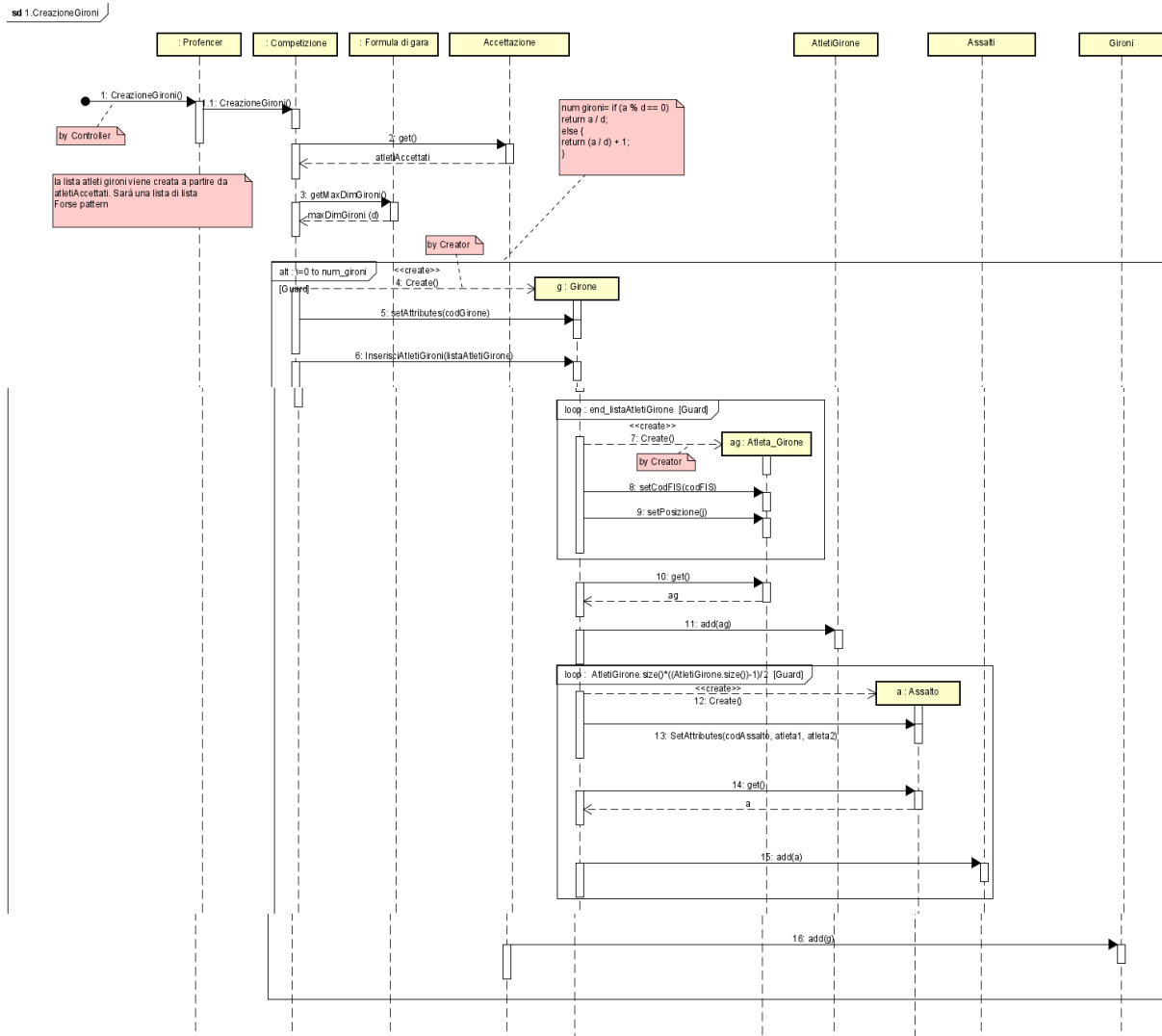
<b>Operazione:</b> InserimentoRisultati(codGirone, listaAssalti)
<b>Riferimenti:</b> Caso d’uso: gestione degli assalti dei gironi
<b>Pre-condizioni:</b> Bisogna avere la lista degli assalti dei gironi con i punteggi
<b>Post-condizioni:</b> i risultati dei gironi sono stati salvati

## Modello di Dominio

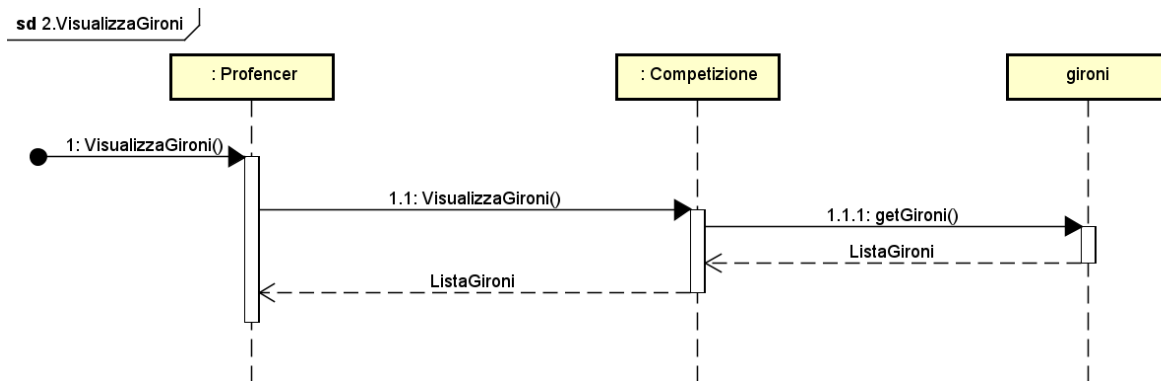


## Diagrammi di Sequenza UC6

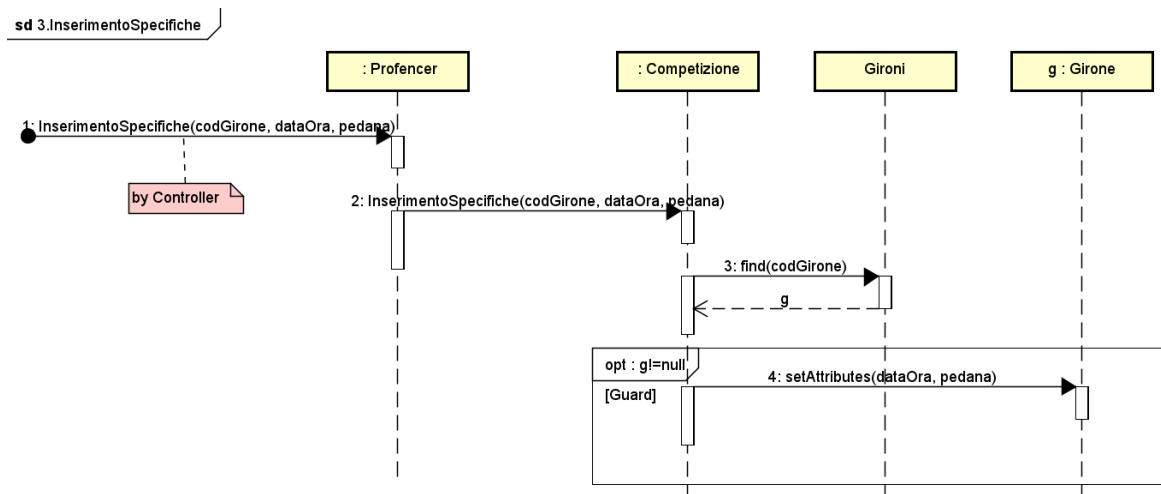
CreazioneGironi



## VisualizzaGironi



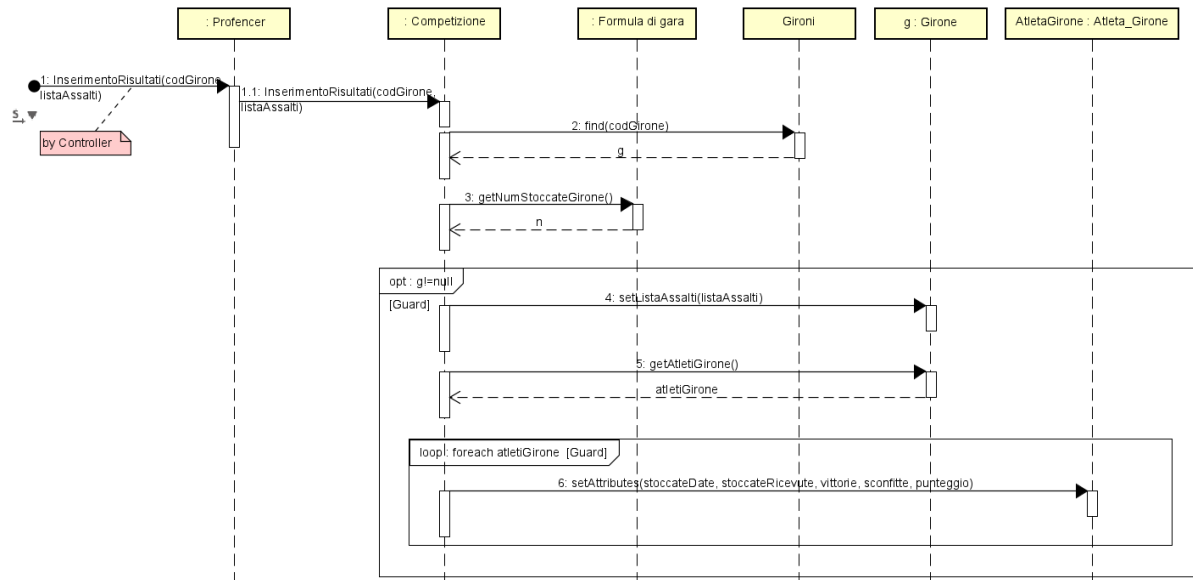
## InserimentoSpecifiche



Diagrammi di Sequenza UC7

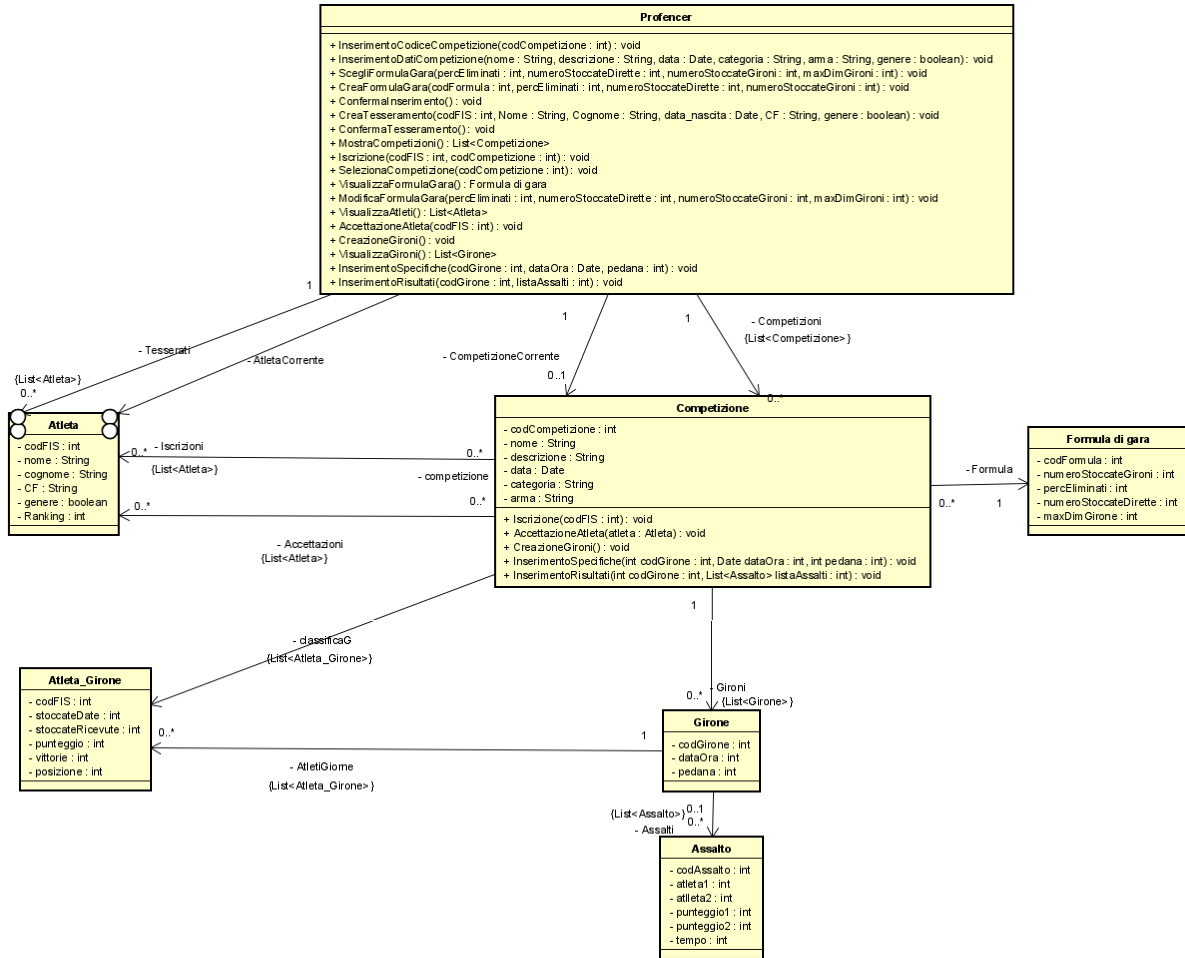
## InserimentoRisultati

sd 2 InserimentoRisultati



# Diagramma delle Classi

pkgModello di Progetto





## Testing

### Testing UC6

```
@Test
public void testCreazioneGironi() {
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        proFencer.CreazioneGironi();
        System.out.println(proFencer.VisualizzaGironi());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}

//Il test funziona correttamente. Viene sollevata un'eccezione se ci sono meno di due atleti accettati

@Test
public void testInserimentoSpecifiche() {
    proFencer.caricaDati();
    try {
        proFencer.CreazioneGironi();
        proFencer.InserimentoSpecifiche(codGirone:1, dataOra:10, pedana:10);
        System.out.println(proFencer.VisualizzaGironi());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
        // TODO: handle exception
    }
}

//Il test funziona correttamente. Viene sollevata un'eccezione se il girone non esiste
```

### Testing UC7

```
@Test
public void testInserimentoRisultati() {
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        proFencer.CreazioneGironi();
        proFencer.InserimentoRisultati(codGirone:1, proFencer.CaricaListaAssalti());
        System.out.println(proFencer.VisualizzaGironi());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}

//Il test funziona correttamente. Viene sollevata un'eccezione se si cercano di inserire risultati non coerenti rispetto la FDG
```

## Iterazione 4

La quarta iterazione implementa gli scenari principali di successo dei casi d'uso:

- UC7.1: Creazione e visualizzazione classifica dopo i gironi
- UC8: Creazione del tabellone ad eliminazione diretta
- UC9: Gestione degli assalti ad eliminazione diretta

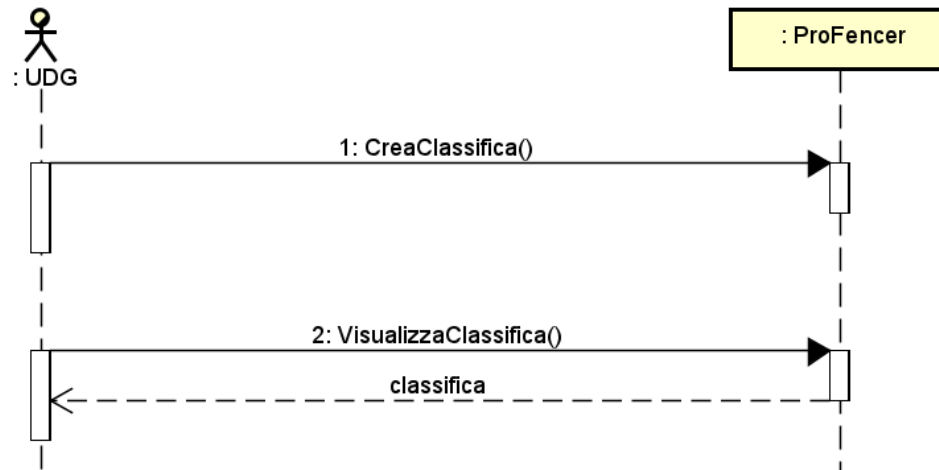
Per ogni caso d'uso vengono effettuate l'analisi, la progettazione, sviluppo del codice e testing.

### UC7.1: Creazione e visualizzazione classifica dopo i gironi

<b>Nome caso d'uso</b>	<b>Creazione e visualizzazione classifica dopo i gironi</b>
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Ufficiale di gara
<b>Parti interessate ed interessi</b>	-L'ufficiale di gara vuole creare e visualizzare la classifica dopo i gironi
<b>Pre-condizioni</b>	I gironi sono stati svolti e i risultati sono stati inseriti
<b>Garanzia di successo</b>	La classifica verrà visualizzata
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"><li>1. L'ufficiale di gara vuole creare la classifica dopo i gironi</li><li>2. L'ufficiale di gara seleziona l'attività 'Classifica Gironi'</li><li>3. ProFencer calcola e visualizza la suddetta classifica</li></ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"><li>1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema</li><li>2. Il Sistema ripristina lo stato</li></ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Una volta per competizione
<b>Varie</b>	

## Diagramma di Sequenza di Sistema

sd SSD\_UC7.1



## Contratti delle Operazioni

### 1. *CreaClassifica*

**Operazione:**

CreaClassifica()

**Riferimenti:**

Caso d'uso: Creazione e visualizzazione classifica dopo i gironi

**Pre-condizioni:**

- Devono essere stati svolti correttamente i gironi

**Post-condizioni:**

- La classifica è stata generata

### 2. *VisualizzaClassifica*

**Operazione:**

VisualizzaClassifica()

**Riferimenti:**

Caso d'uso: Creazione e visualizzazione classifica dopo i gironi

**Pre-condizioni:**

- La classifica è stata generata

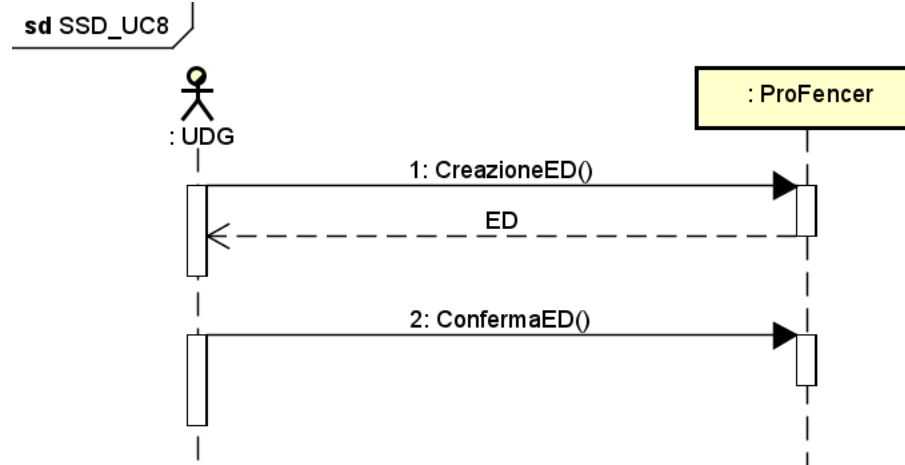
**Post-condizioni:**

- La classifica viene visualizzata

## UC8: Creazione del tabellone ad eliminazione diretta

<b>Nome caso d'uso</b>	<b>Creazione del tabellone di eliminazione diretta</b>
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Ufficiale di gara
<b>Parti interessate ed interessi</b>	-L'ufficiale di gara vuole generare il tabellone ad eliminazione diretta
<b>Pre-condizioni</b>	Tutti i gironi sono stati svolti ed è stata creata la classifica dopo i gironi
<b>Garanzia di successo</b>	Si crea la prima fase dell'eliminazione diretta
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"> <li>1. L'ufficiale di gara seleziona l'attività crea tabellone di eliminazione diretta.</li> <li>2. Visualizza il tabellone generato</li> <li>3. Conferma l'operazione</li> </ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"> <li>1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema</li> <li>2. Il Sistema ripristina lo stato</li> </ol> <p>1a. L'ufficiale di gara seleziona l'attività quando ancora i gironi non sono stati svolti</p> <ol style="list-style-type: none"> <li>1. Il sistema genera un messaggio di errore</li> </ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Una volta per competizione
<b>Varie</b>	

## Diagramma di Sequenza di Sistema



## Contratti delle Operazioni

### 1. Creazione Tabellone ad Eliminazione Diretta

**Operazione:**

CreazioneED()

**Riferimenti:**

Caso d'uso: Creazione del tabellone di ED

**Pre-condizioni:**

- Tutti i gironi devono essere compilati
- Deve essere stata selezionata la competizione corrente

**Post-condizioni:**

- Viene creata l'entità ED con tutti gli assalti ancora non compilati
- Vengono visualizzati
- ed diventa corrente

### 2. Conferma Tabellone ad Eliminazione Diretta

**Operazione:**

ConfermaED()

**Riferimenti:**

Caso d'uso: Creazione del tabellone di ED

**Pre-condizioni:** il tabellone di ED è stato creato

**Post-condizioni:**

- Si salva la ed corrente nella lista

## UC9: Gestione degli assalti ad eliminazione diretta

<b>Nome caso d'uso</b>	<b>Gestione degli assalti ad eliminazione diretta</b>
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Ufficiale di gara
<b>Parti interessate ed interessi</b>	-L'ufficiale di gara vuole inserire i risultati del tabellone di eliminazione diretta corrente
<b>Pre-condizioni</b>	Il tabellone di ED è stato creato
<b>Garanzia di successo</b>	Il risultato è stato salvato
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"> <li>1. L'ufficiale di gara seleziona l'attività 'Gestione Assalti ED'.</li> <li>2. L'Ufficiale di Gara seleziona l'ED da gestire e questo diventa corrente</li> <li>3. Si visualizza il tabellone ED corrente</li> <li>4. Si compilano uno o più assalti</li> <li>5. Si invia il risultato degli assalti</li> <li>6. Viene aggiornato il tabellone di ED</li> <li>7. Se tutti gli assalti dell'ED corrente sono stati compilati si crea il tabellone successivo</li> </ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"> <li>1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema</li> <li>2. Il Sistema ripristina lo stato</li> </ol> <p>1a. L'ufficiale di gara seleziona l'attività quando ancora non è stato creato il tabellone di ED.</p> <ol style="list-style-type: none"> <li>1. Il sistema genera un messaggio di errore</li> </ol> <p>5a. L'UDG inserisce punteggi non validi</p> <ol style="list-style-type: none"> <li>1. Il sistema genera un messaggio di errore</li> <li>2. Si torna allo stato 3</li> </ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Dipende dal numero di assalti di ED
<b>Varie</b>	

## Diagramma di Sequenza di Sistema

sd SSD\_UC9



## Contratti delle Operazioni

### 1. Selezione del Tabellone di Eliminazione Diretta

**Operazione:**

SelezionaED(stato)

**Riferimenti:**

Caso d'uso: Gestione degli assalti ad eliminazione diretta

**Pre-condizioni:**

- Deve esistere almeno un tabellone

**Post-condizioni:**

- ed diventa corrente

### 2. Visualizza Tabellone ED corrente

**Operazione:**

VisualizzaED()

**Riferimenti:**

Caso d'uso: Gestione degli assalti ad eliminazione diretta

**Pre-condizioni:**

- Esiste un ed corrente

**Post-condizioni:**

- Si visualizza il tabellone

### 3. *Inserimento Risultati ED*

**Operazione:**

InserisciRisultatiED(listaAssalti)

**Riferimenti:**

Caso d'uso: Gestione degli assalti ad eliminazione diretta

**Pre-condizioni:**

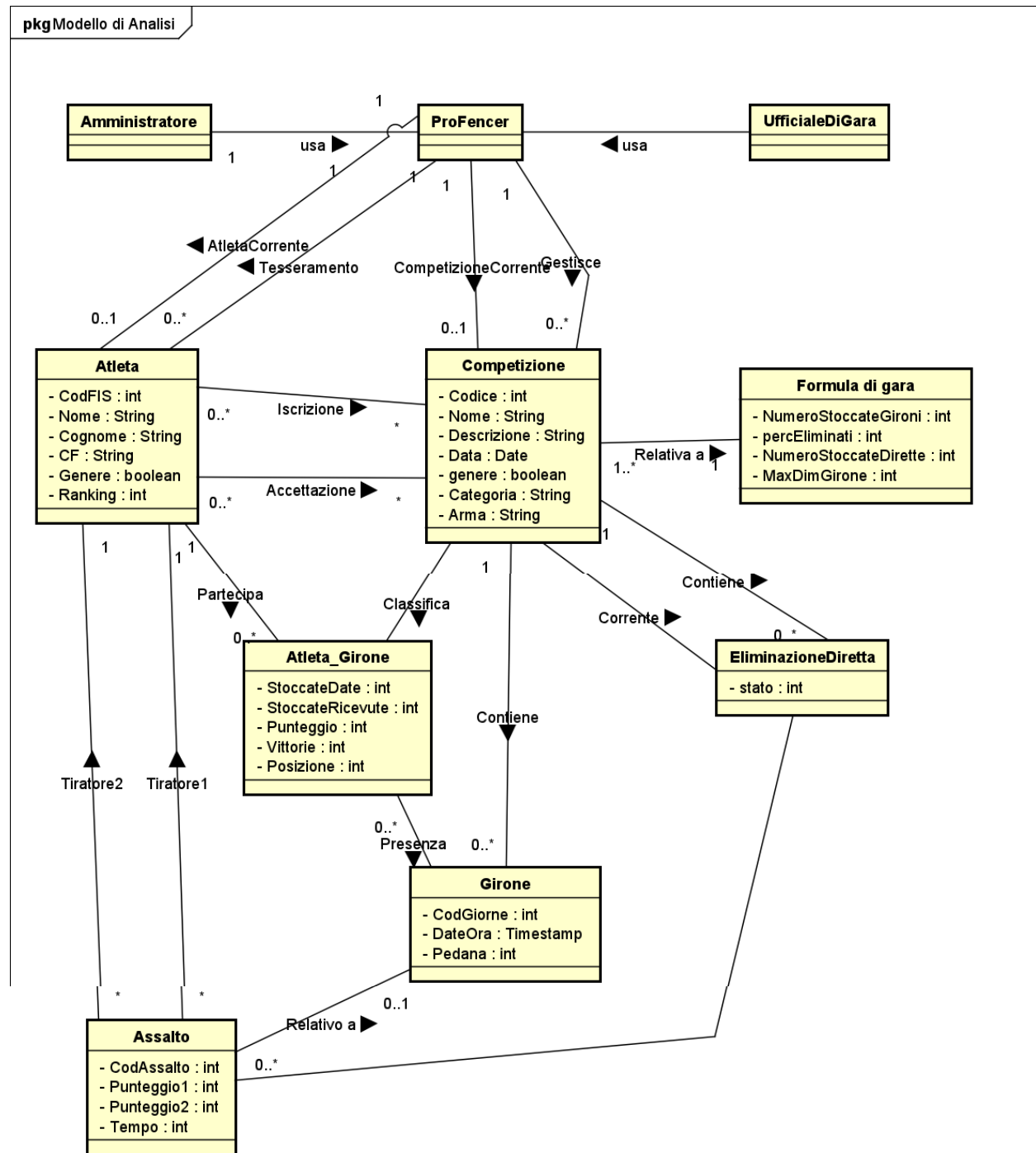
- ed è corrente

**Post-condizioni:**

- ed modificata viene aggiunta alla lista eliminazioniDirette



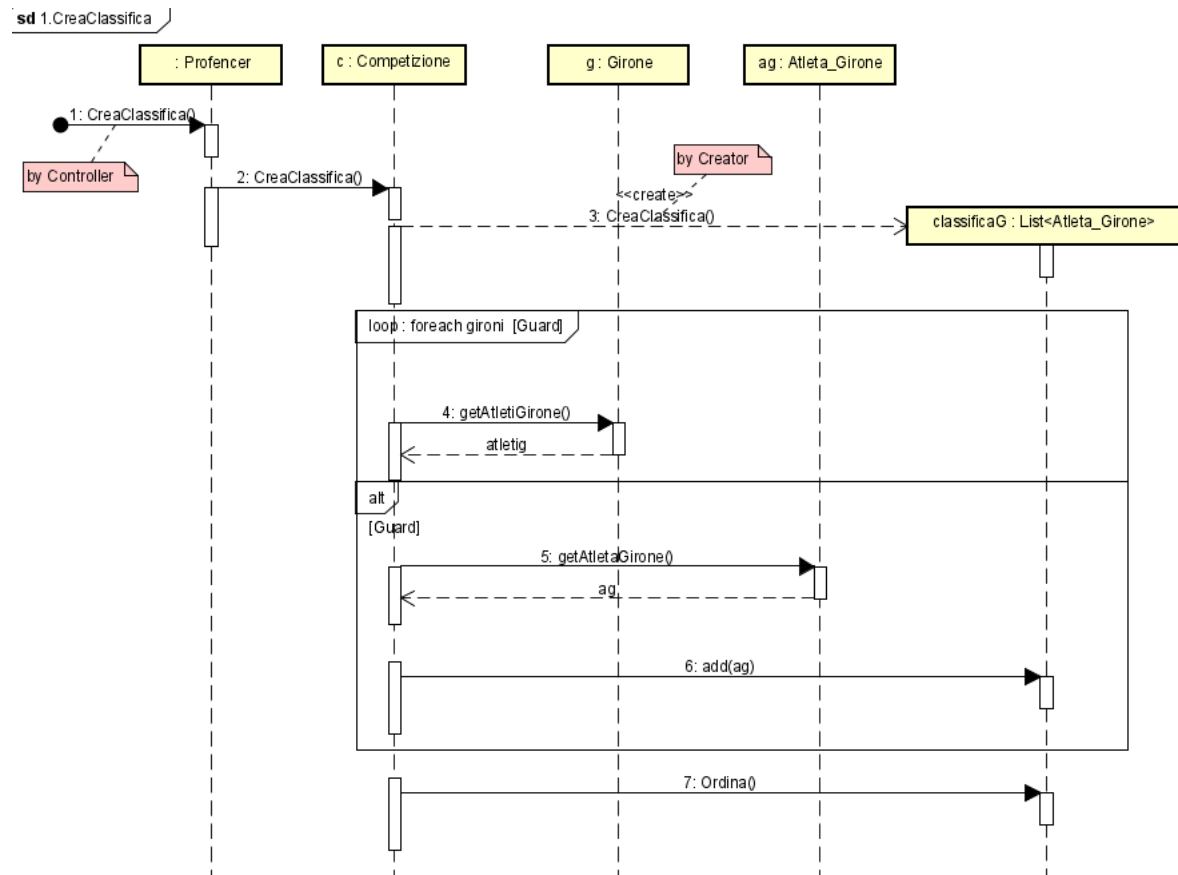
## Modello di Dominio



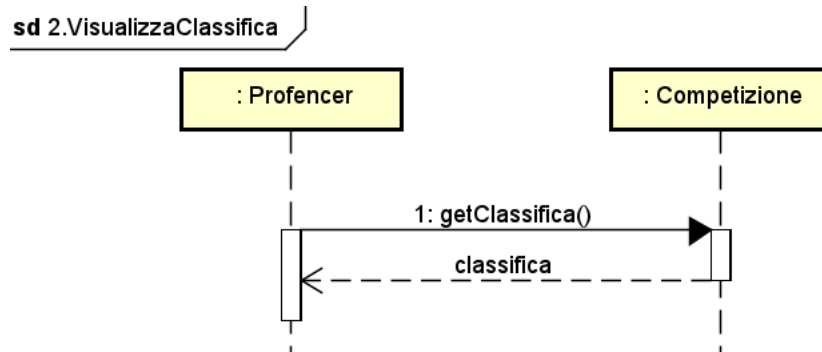
## Progettazione

### Diagrammi di Sequenza UC7.1

#### CreaClassifica

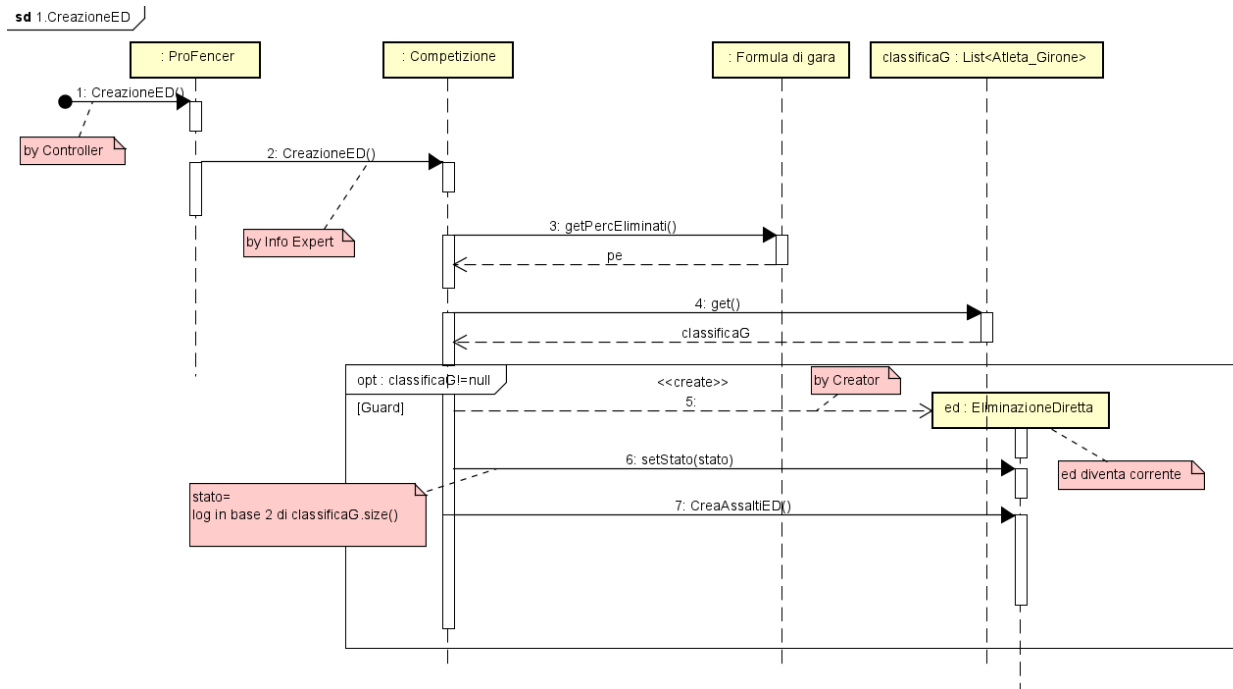


#### VisualizzaClassifica

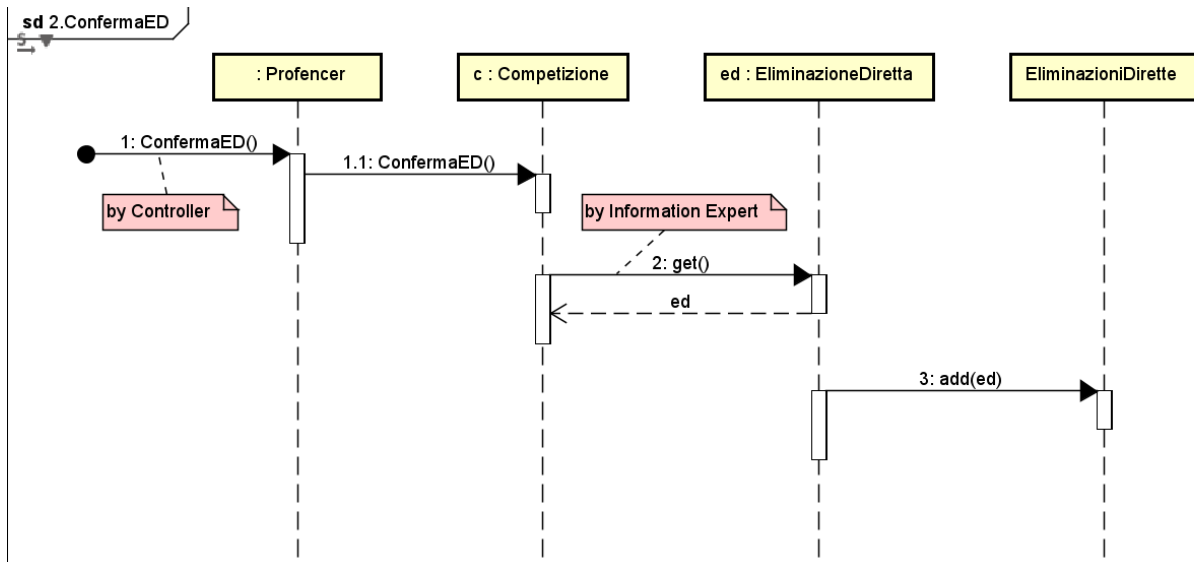


## Diagrammi di Sequenza UC8

### CreazioneED

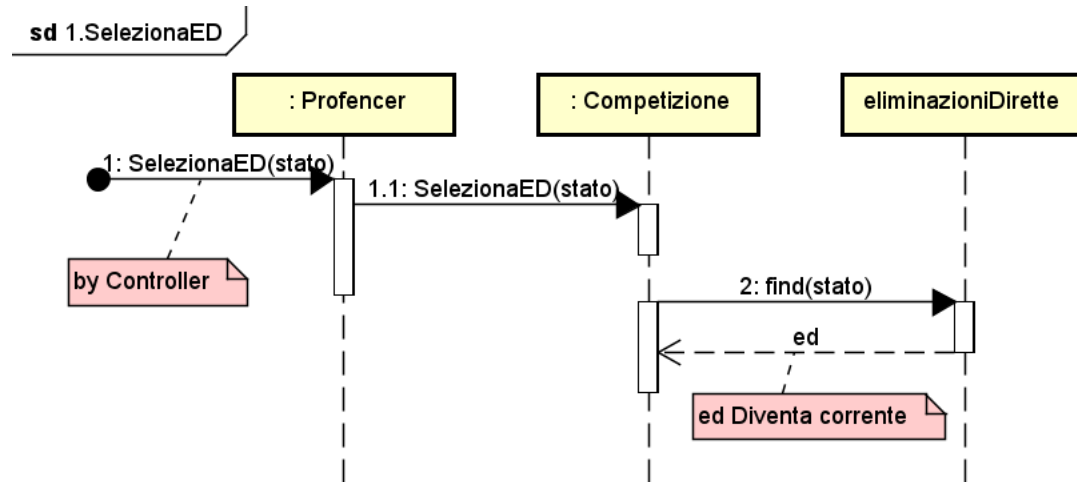


### ConfermaED

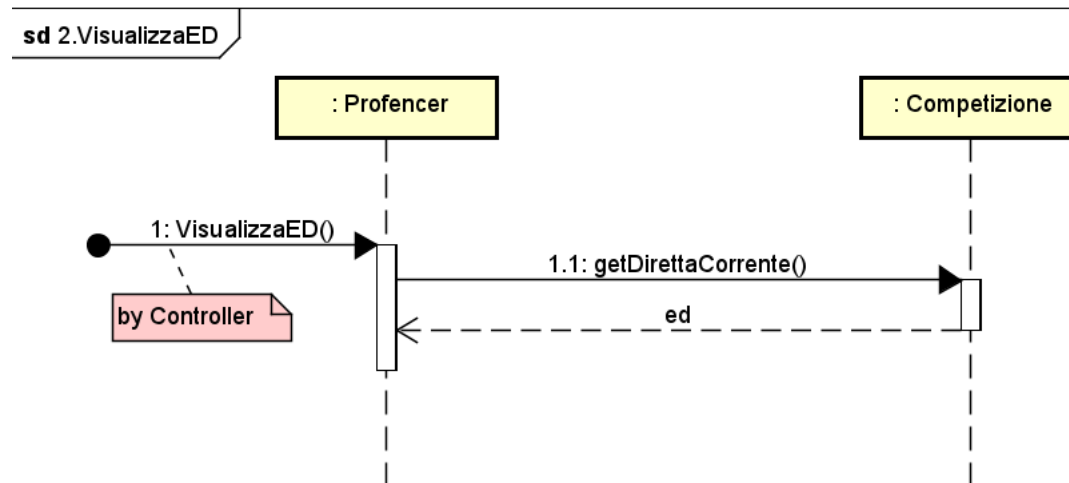


## Diagrammi di Sequenza UC9

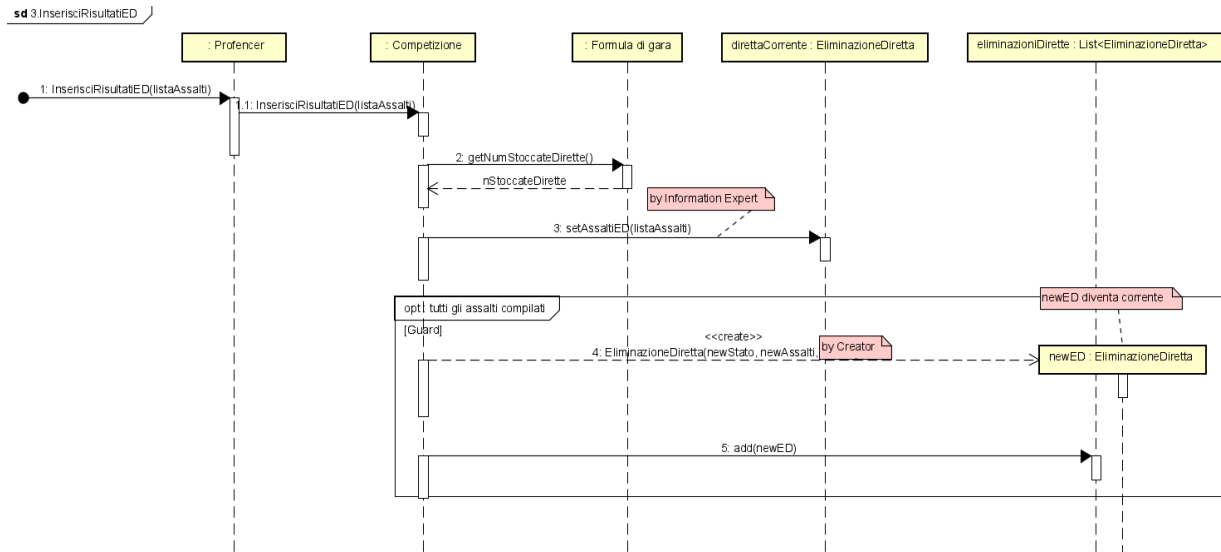
### SelezionaED



### VisualizzaED

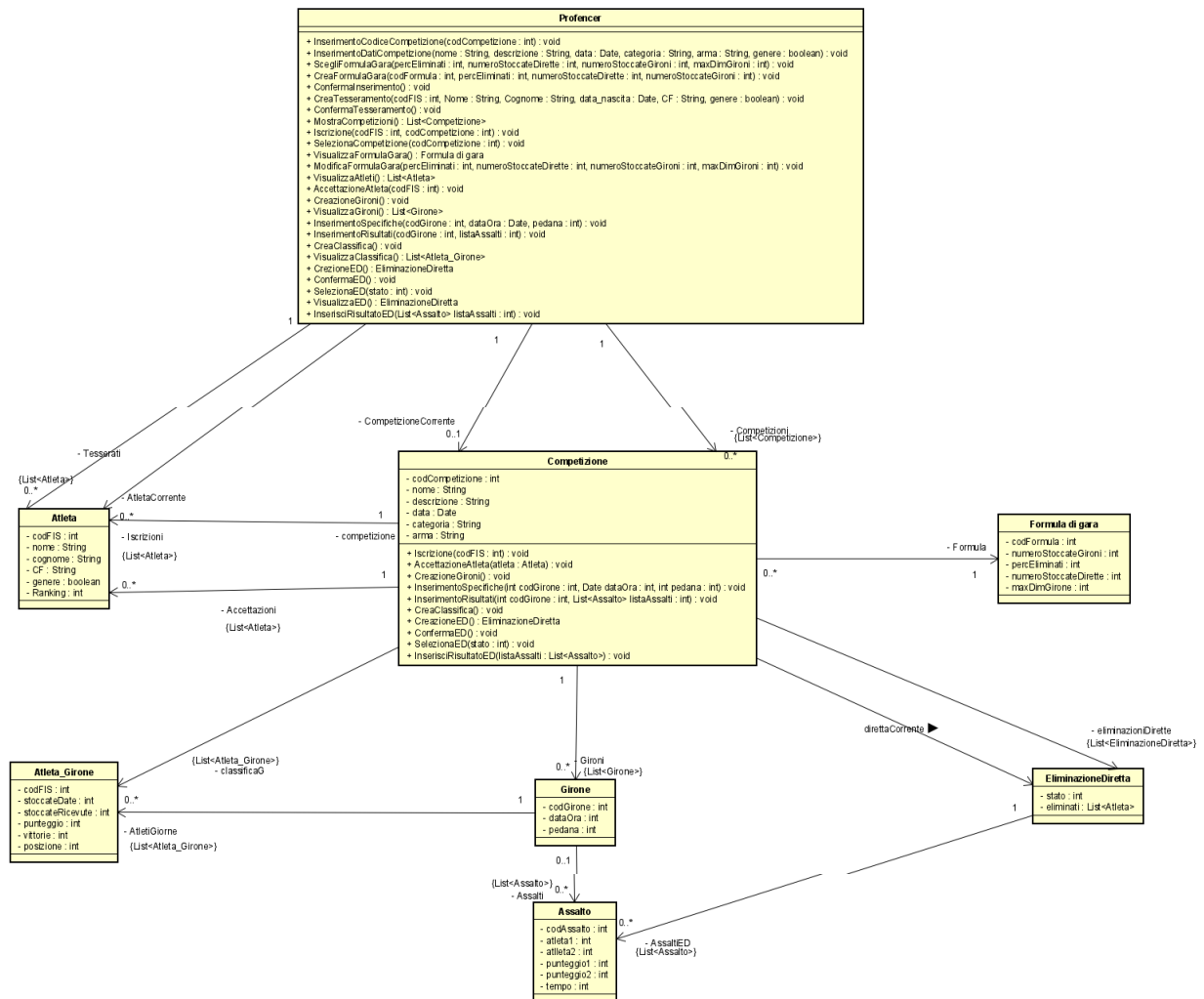


### InserimentoRisultatiED



# Diagramma delle Classi

pkg Modello di Progetto



## Testing

### Testing UC7.1

```
@Test
public void CreaClassifica(){
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        proFencer.CreazioneGironi();
        proFencer.InserimentoRisultati(codGirone:1, proFencer.CaricaListaAssalti());
        proFencer.CreaClassifica();
        System.out.println(proFencer.VisualizzaClassifica());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
        // TODO: handle exception
    }
}
//Il test funzione correttamente. Viene sollevata un'eccezione se ancora non esistono gironi
```

### Testing UC8

```
@Test
public void testCreazioneED() {
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        proFencer.CreazioneGironi();
        proFencer.InserimentoRisultati(codGirone:1, proFencer.CaricaListaAssalti());
        proFencer.CreaClassifica();
        System.out.println(proFencer.CreazioneED());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}
//Il test funzione correttamente. Viene sollevata un'eccezione se ancora non è stata creata la classifica dopo i gironi

@Test
public void testConfermaED() {
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        proFencer.CreazioneGironi();
        proFencer.InserimentoRisultati(codGirone:1, proFencer.CaricaListaAssalti());
        proFencer.CreaClassifica();
        proFencer.CreazioneED();
        System.out.println(proFencer.getCompetizioneCorrente().getDirettaCorrente());
        proFencer.ConfermaED();
        System.out.println(proFencer.getCompetizioneCorrente().getEliminazioniDirette());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}
//Il test funzione correttamente. Viene sollevata un'eccezione se non esiste una diretta corrente
```

## Testing UC9

```
@Test
public void testSelezionaED(){
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        proFencer.CreazioneGironi();
        proFencer.InserimentoRisultati(codGirone:1, proFencer.CaricaListaAssalti());
        proFencer.CreaClassifica();
        proFencer.CreazioneED();
        proFencer.getCompetizioneCorrente();
        proFencer.ConfermaED();
        proFencer.SelezionaED(stato:3);
        System.out.println(proFencer.getCompetizioneCorrente().getDirettaCorrente());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
        // TODO: handle exception
    }
}

//Il test funzione correttamente. Viene sollevata un'eccezione se si seleziona un'ED non esistente

@Test
public void testInserisciRisultatiED() {
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        proFencer.CreazioneGironi();
        proFencer.InserimentoRisultati(codGirone:1, proFencer.CaricaListaAssalti());
        proFencer.CreaClassifica();
        proFencer.CreazioneED();
        proFencer.getCompetizioneCorrente();
        proFencer.ConfermaED();
        proFencer.SelezionaED(stato:3);
        proFencer.InserisciRisultatiED(proFencer.CaricaListaAssaltiED());
        System.out.println(proFencer.VisualizzaED());
        System.out.println(proFencer.getCompetizioneCorrente().getEliminazioniDirette());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}

//Il test funzione correttamente. Viene sollevata un'eccezione se si inseriscono punteggio non coerenti con la FDG
```



## Iterazione 5

La quinta e ultima iterazione implementa gli scenari principali di successo dei casi d'uso:

- UC10: Generazione del ranking
- UC11: Visualizzazione delle classifiche

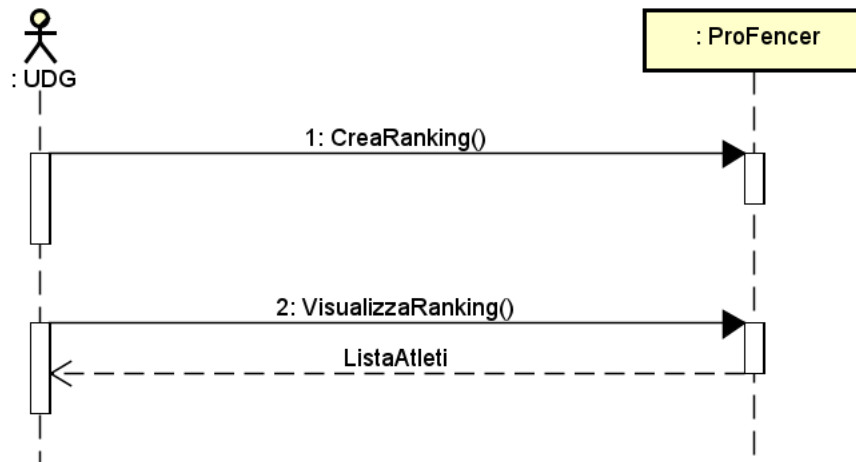
Per ogni caso d'uso vengono effettuate l'analisi, la progettazione, sviluppo del codice e testing.

### UC10: Generazione del ranking

<b>Nome caso d'uso</b>	<b>Gestione degli assalti ad eliminazione diretta</b>
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Ufficiale di gara
<b>Parti interessate ed interessi</b>	-L'ufficiale di gara vuole calcolare il ranking quando la competizione è terminata
<b>Pre-condizioni</b>	La competizione si è conclusa (sono stati svolti tutti gli assalti ad ED)
<b>Garanzia di successo</b>	Ogni atleta facente parte della competizione appena conclusa avrà un nuovo punteggio nel ranking
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"><li>1. L'ufficiale di gara seleziona l'attività 'Crea ranking'.</li><li>2. Il sistema calcola il ranking</li><li>3. Il ranking viene visualizzato</li></ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"><li>1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema</li><li>2. Il Sistema ripristina lo stato</li></ol> <p>1a. L'UDG prova a creare il ranking più di una volta</p> <ol style="list-style-type: none"><li>1. Il sistema non ricalcola il ranking</li><li>2. Il sistema visualizza il ranking esistente</li></ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	Abbiamo utilizzato l'attuale tabella federale del ranking per i dati necessari al ricalcolo del ranking
<b>Frequenza di ripetizioni</b>	Una volta per competizione
<b>Varie</b>	

## Diagramma di Sequenza del Sistema

sd SSD\_UC10



## Contratti delle Operazioni

### 1. Crea Ranking

**Operazione:**

CreaRanking()

**Riferimenti:**

Caso d'uso: Generazione dei ranking

**Pre-condizioni:**

- La gara deve essere stata completata

**Post-condizioni:**

- Gli atleti partecipanti alla competizione avranno un nuovo punteggio ranking

### 2. Visualizza Ranking

**Operazione:**

VisualizzaRanking()

**Riferimenti:**

Caso d'uso: Generazione dei ranking

**Pre-condizioni:****Post-condizioni:**

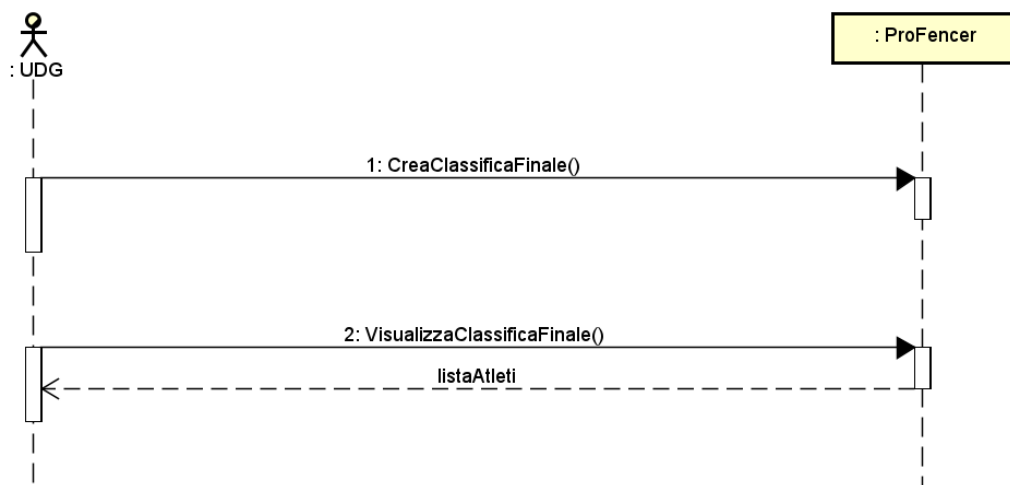
- Viene visualizzato il ranking

## UC11: Visualizzazione delle classifiche

<b>Nome caso d'uso</b>	<b>Gestione degli assalti ad eliminazione diretta</b>
<b>Portata</b>	Applicazione ProFencer
<b>Livello</b>	Obiettivo utente
<b>Attore primario</b>	Ufficiale di gara
<b>Parti interessate ed interessi</b>	-L'ufficiale di gara vuole stampare la classifica della competizione appena disputata
<b>Pre-condizioni</b>	La competizione si è conclusa (sono stati svolti tutti gli assalti ad ED)
<b>Garanzia di successo</b>	Si avrà una classifica degli atleti
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"> <li>1. L'ufficiale di gara seleziona l'attività 'Stampa classifica finale'.</li> <li>2. Il sistema calcola la classifica finale</li> <li>3. La classifica viene visualizzata dal sistema</li> </ol>
<b>Estensioni</b>	<p>a. In un qualsiasi momento il Sistema fallisce e si arresta improvvisamente.</p> <ol style="list-style-type: none"> <li>1. L'Amministratore riavvia il software e ripristina lo stato precedente del Sistema</li> <li>2. Il Sistema ripristina lo stato</li> </ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	
<b>Frequenza di ripetizioni</b>	Una volta per competizione
<b>Varie</b>	

## Diagramma di Sequenza di Sistema

sd SSD\_UC11



## Contratti delle Operazioni

### 1. *CreaClassificaFinale*

**Operazione:**

CreaClassificaFinale()

**Riferimenti:**

Caso d'uso: Visualizzazione delle classifiche

**Pre-condizioni:**

- La competizione deve essere stata completata

**Post-condizioni:**

- La classifica finale viene salvata

### 2. *VisualizzaClassificaFinale*

**Operazione:**

VisualizzaClassificaFinale()

**Riferimenti:**

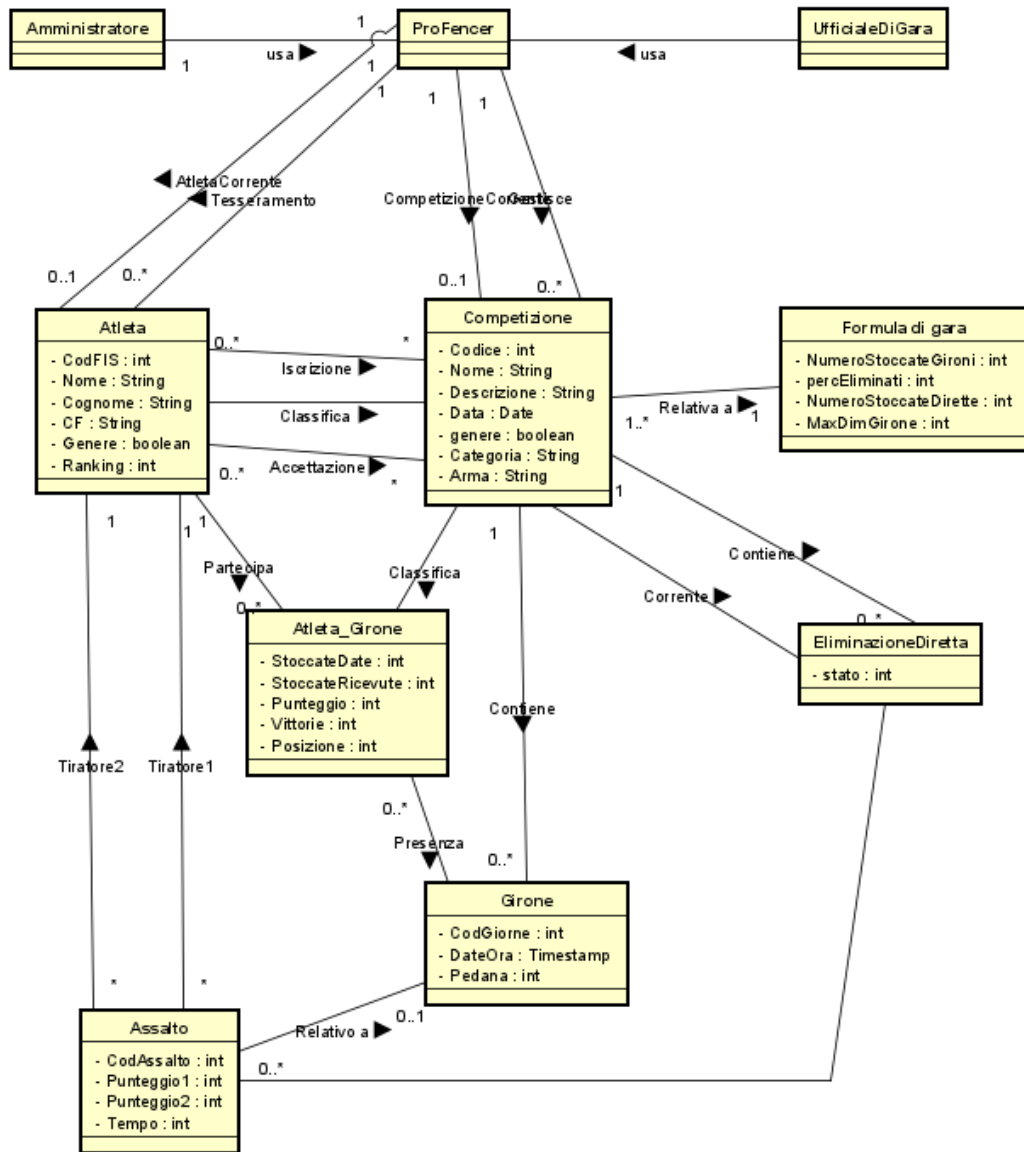
Caso d'uso: Visualizzazione delle classifiche

**Pre-condizioni:****Post-condizioni:**

- Viene visualizzata la classifica finale

## Modello di Dominio

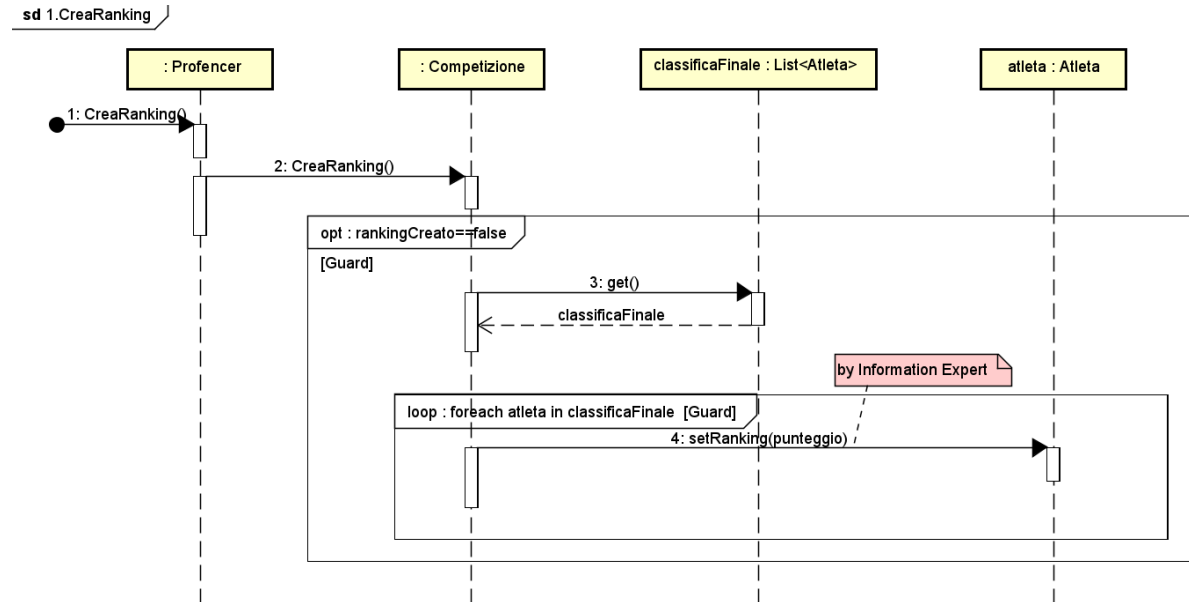
pkg Modello di Analisi



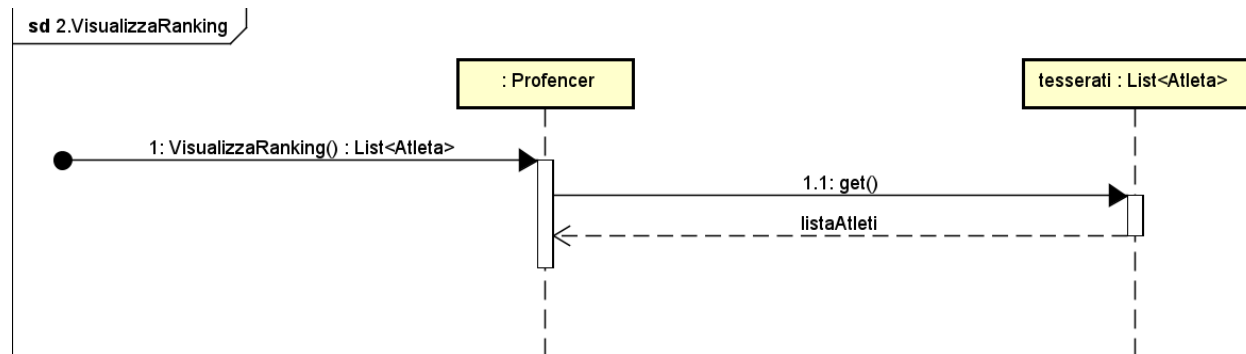
## Progettazione

### Diagrammi di Sequenza UC10

#### CreaRanking

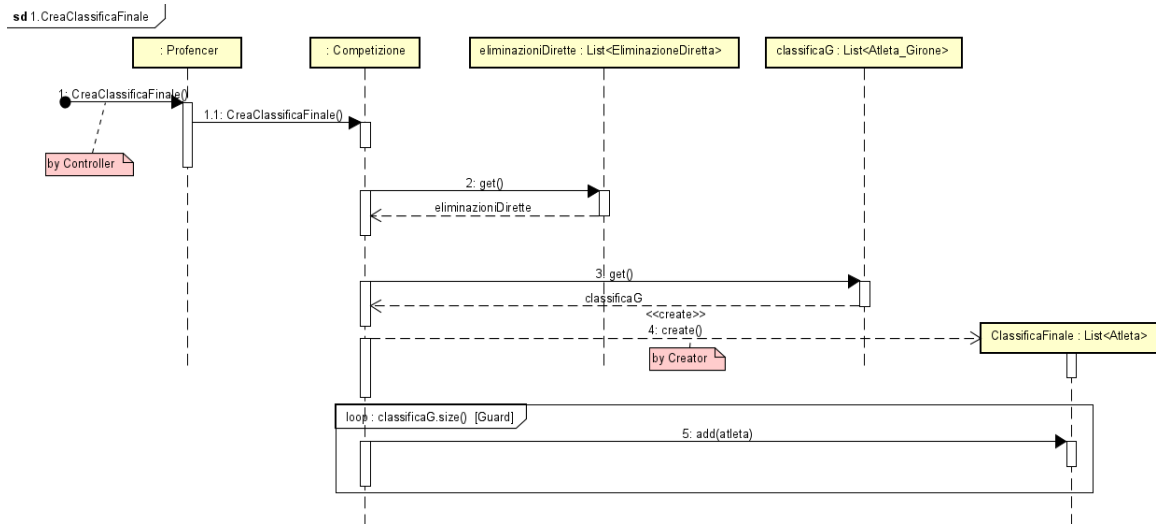


#### VisualizzaRanking

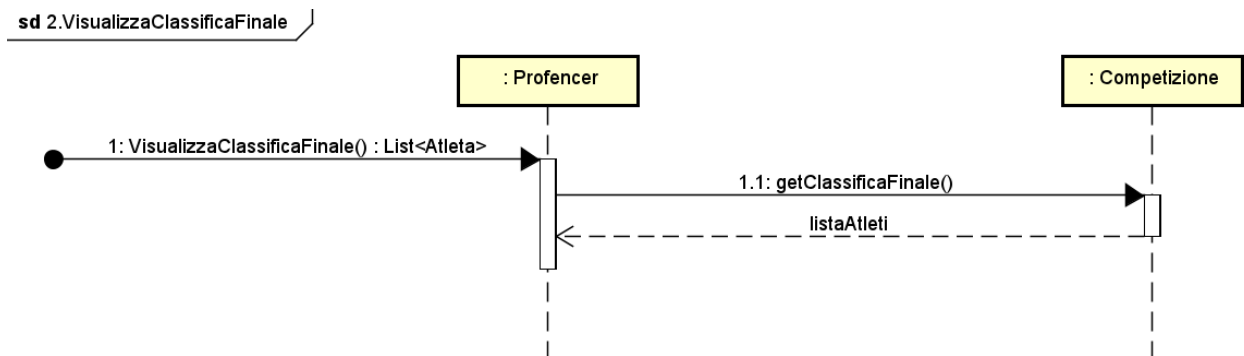


## Diagrammi di Sequenza UC11

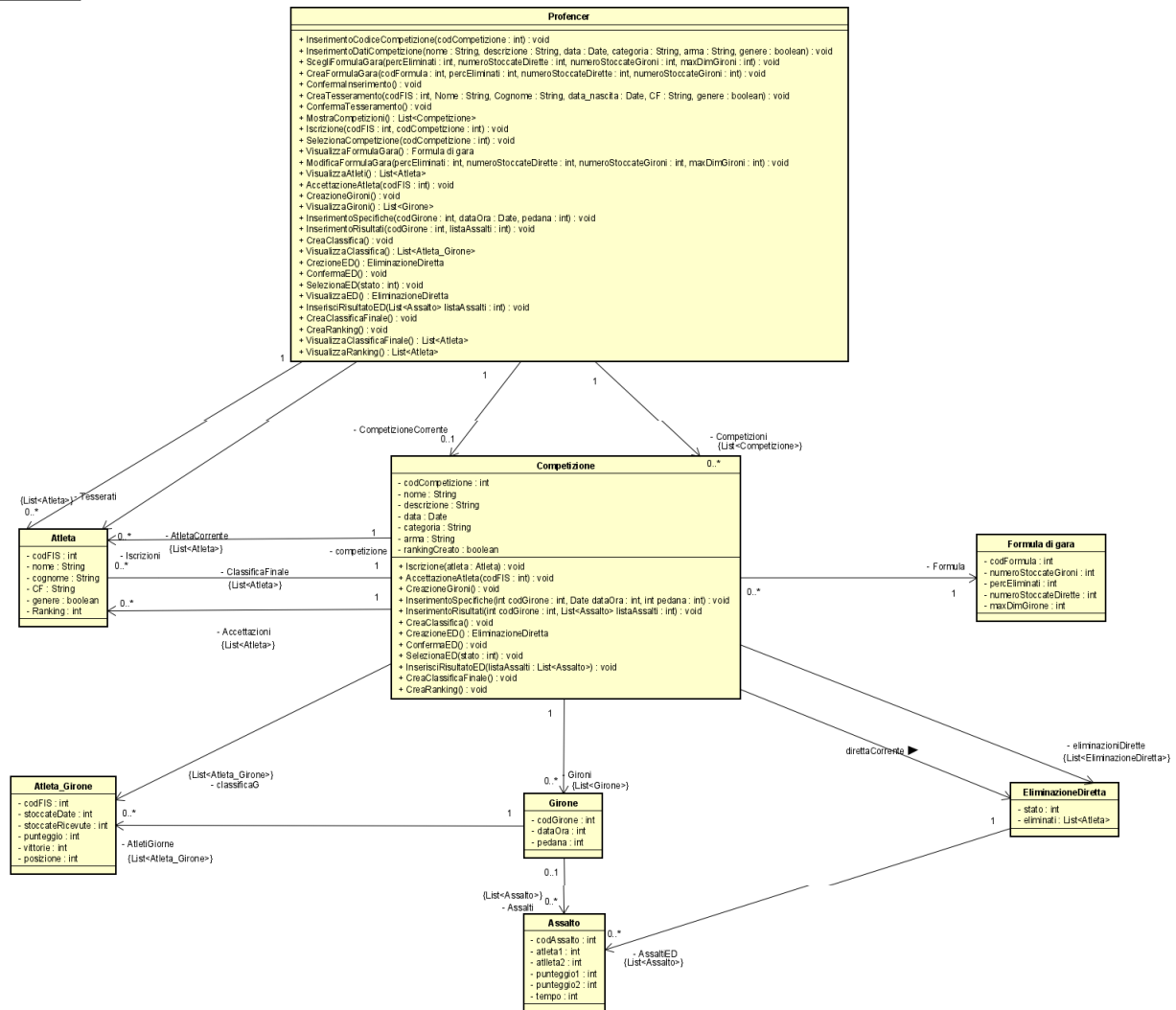
### CreaClassificaFinale



### VisualizzaClassificaFinale



## Diagramma delle Classi





## Testing

### Testing UC10

```
@Test
public void testCreaClassificaFinale() {
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        proFencer.CreazioneGironi();
        proFencer.InserimentoRisultati(codGirone:1, proFencer.CaricaListaAssalti());
        proFencer.CreaClassifica();
        //System.out.println(proFencer.getCompetizioneCorrente().getClassificaG());
        proFencer.CreazioneED();
        proFencer.getCompetizioneCorrente();
        proFencer.ConfermaED();
        proFencer.SelezionaED(stato:3);
        proFencer.InserisciRisultatiED(proFencer.CaricaListaAssaltiED());
        proFencer.SelezionaED(stato:2);
        proFencer.InserisciRisultatiED(proFencer.CaricaListaAssaltiED3());
        proFencer.SelezionaED(stato:1);
        proFencer.InserisciRisultatiED(proFencer.CaricaListaAssaltiED4());
        //System.out.println(proFencer.getCompetizioneCorrente());
        proFencer.CreaClassificaFinale();
        System.out.println(proFencer.getCompetizioneCorrente().getClassificaFinale());
        System.out.println((proFencer.getCompetizioneCorrente().getClassificaFinale().size()));

    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}

//Il test funzione correttamente. Viene sollevata un'eccezione se la gara non è stata completata
```

## Testing UC11

```
@Test
public void testCreaRanking() {
    proFencer.caricaDati();
    try {
        proFencer.SelezionaCompetizione(codCompetizione:1);
        proFencer.CreazioneGironi();
        proFencer.InserimentoRisultati(codGirone:1, proFencer.CaricaListaAssalti());
        proFencer.CreaClassifica();
        //System.out.println(proFencer.getCompetizioneCorrente().getClassificaG());
        proFencer.CreazioneED();
        proFencer.getCompetizioneCorrente();
        proFencer.ConfermaED();
        proFencer.SelezionaED(stato:3);
        proFencer.InserisciRisultatiED(proFencer.CaricaListaAssaltiED());
        proFencer.SelezionaED(stato:2);
        proFencer.InserisciRisultatiED(proFencer.CaricaListaAssaltiED3());
        proFencer.SelezionaED(stato:1);
        proFencer.InserisciRisultatiED(proFencer.CaricaListaAssaltiED4());
        //System.out.println(proFencer.getCompetizioneCorrente());
        proFencer.CreaRanking();
        System.out.println(proFencer.getTesserati());
    } catch (Exception e) {
        System.out.println(e.getMessage());
        fail();
    }
}

//Il test funzione correttamente. Viene sollevata un'eccezione se la gara non è stata completata
```

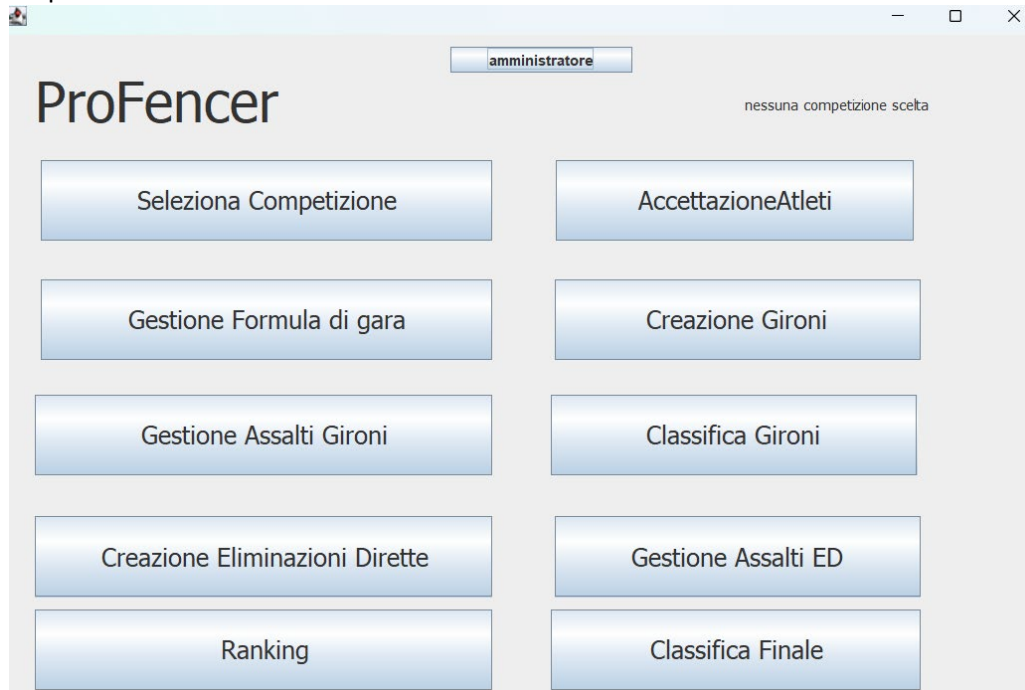
## Interfaccia grafica

Tutto il software è fruibile attraverso l'utilizzo di un'interfaccia grafica, questa è stata sviluppata gradualmente insieme alle altre componenti del software.

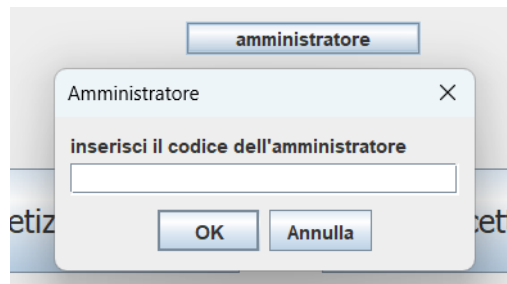
La GUI ha diverse schermate per ogni utente che interagisce con il software, cioè l'amministratore e l'ufficiale di gara.

Tutte le schermate sollevano eccezioni e li notificano agli utenti quando necessario

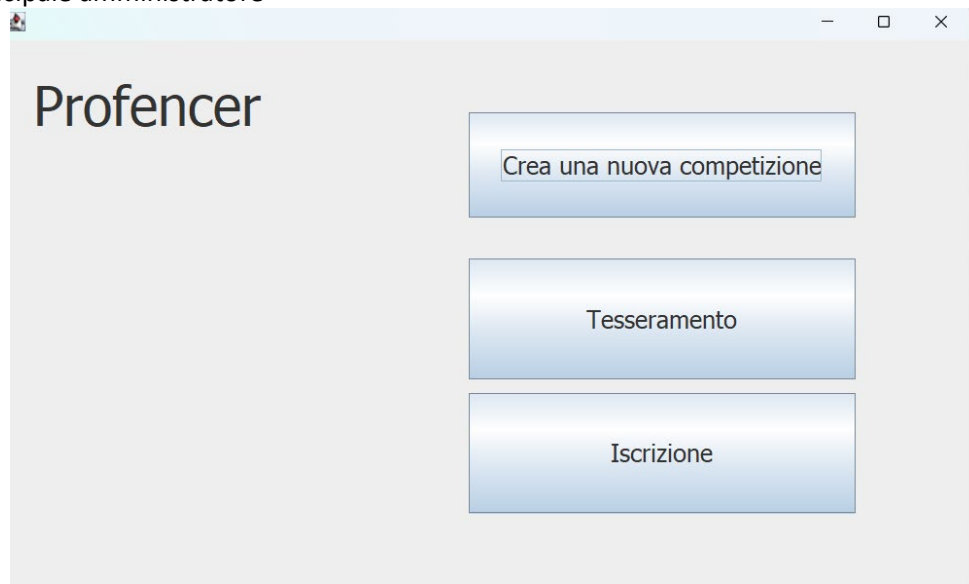
Frame principale di UDG



Per passare alla schermata dell'amministratore è necessario inserire un codice



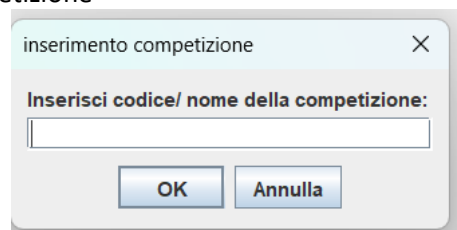
Frame principale amministratore



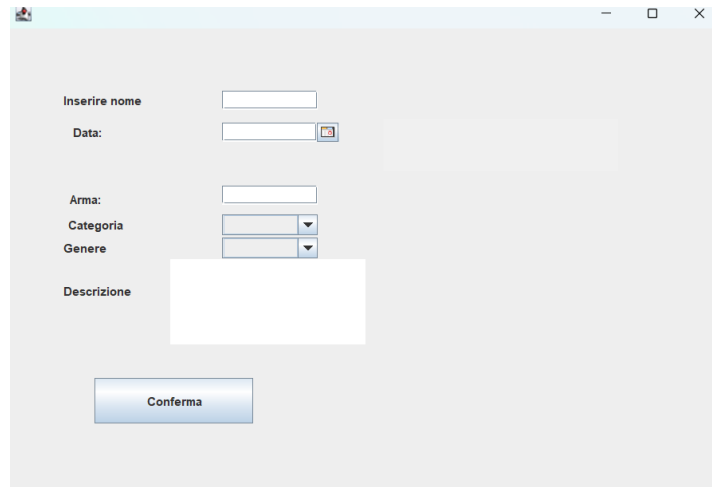
Nelle schermate dell'interfaccia grafica sono presenti pulsanti che consentono di eseguire tutte le azioni previste nei vari casi d'uso del software.

## UC1

Inserimento codice nuova competizione



Creazione della competizione



Input della formula di gara

The image shows four stacked dialog boxes, each titled "specifiche" with a close button (X) in the top right corner. Each dialog box contains a label, an input field, and two buttons labeled "OK" and "Annulla".

- Dialog 1: Label "Inserisci numero di stoccate per i gironi :", empty input field.
- Dialog 2: Label "Inserisci percentuale di eliminati in formato intero:", empty input field.
- Dialog 3: Label "Inserisci numero di stoccate per la fase ad eliminazione:", empty input field.
- Dialog 4: Label "Inserisci il numero massimo di atleti:", empty input field.

UC2

Tesseramento di un atleta

The image shows a window titled "Tesseramento" with a close button (X) in the top right corner. The window contains several input fields and a "Conferma" button.

**Tesseramento**

Codice FIS:

Nome dell'atleta:

Cognome dell'atleta:

CF:

genere:

Data di nascita:

### UC3

#### Iscrizione di un'atleta

The interface displays a list of competitions on the left and a corresponding list of 'Iscriviti alla gara' (Register for the race) buttons on the right. A modal dialog titled 'iscrizione' is open, prompting the user to 'Inserisci il codice fis dell'atleta :'. The modal includes an input field and 'OK' and 'Annulla' (Cancel) buttons.

Competizione:	Button
1	Iscriviti alla gara
2	Iscriviti alla gara
3	Iscriviti alla gara
4	Iscriviti alla gara
10	Iscriviti alla gara

Modal Dialog: iscrizione

Inserisci il codice fis dell'atleta :

OK Annulla

### UC4

#### Accettazione degli atleti

The interface displays a list of athlete names on the left and a corresponding list of 'Accetta atleta' (Accept athlete) buttons on the right.

Nome Atleta	Button
Luigi Bianchi	Accetta atleta
Leandro Boro	Accetta atleta
Lucia Boro	Accetta atleta
Luisa Ricci	Accetta atleta
Carla Privitera	Accetta atleta
Carlo Bialilli	Accetta atleta

### UC5

#### Modifica formula di gara

The interface displays the title 'gestione della formula di gara della competizione : 2'. Below the title, there are four input fields for the following parameters:

- Numero Stoccate gironi: 5
- Numero massimo atleti in un giorno : 5
- Percentuale eliminati: 0
- Numero Stoccate eliminazioni diretta: 10

A 'Conferma Modifiche' (Confirm Changes) button is located at the bottom right of the form.

## UC6

### Visualizza gironi

The window displays two columns for tournament rounds. Each column has a header 'girone: 1' and 'girone: 2' respectively, followed by an 'inserisci dati' button. Below each button is a list of athlete codes.

girone: 1	girone: 2
Codice Atleta1:40000	Codice Atleta1:2
Codice Atleta1:40000	Codice Atleta2:3
Codice Atleta1:67890	Codice Atleta1:2
Codice Atleta2:4	Codice Atleta2:5
Codice Atleta2:4	Codice Atleta1:3
	Codice Atleta2:5

## UC7

### Gestione assalti dei gironi

The window displays a table with three columns. The first column contains athlete codes (40000, 67890), the second column contains scores (2, 4, 0), and the third column contains scores (1, 2, 0). A 'Conferma' button is at the bottom.

40000	67890	2	4	0
40000	4	1	2	0
67890	4	1	5	0

## UC7.1

### Visualizza classifica dopo i gironi

The window displays a table titled 'Classifica Gironi' with the following columns: atleta, Stoccate Date, Staccate Ricev., Punteggio, and Vittorie.

atleta	Stoccate Date	Staccate Ricev..	Punteggio	Vittorie
4	7	2	1.0	2
3	6	2	1.0	2
5	5	4	0.5	1
67890	5	7	0.5	1
40000	3	6	0.0	0
2	1	6	0.0	0

## UC8

### Creazione tabellone di eliminazione diretta

Tabellone eliminazione diretta		
4		0
3		0
5	2	0
67890	40000	0

Conferma

Il -2 è un valore utile al software per comprendere se l'assalto è stato disputato o meno, -2 ci dice che l'assalto deve essere ancora disputato, un valore  $\geq 0$  ci dice il risultato dell'assalto. In aggiornamenti futuri è possibile sostituire questo valore graficamente con, ad esempio, una stringa che segnala all'utente che l'assalto è ancora da disputare.

## UC9

### Seleziona tabellone di eliminazione diretta.

Tabellone dei 8  
Tabellone dei 4  
FINALE

Seleziona fase

### Gestione degli assalti di eliminazione diretta

Gestione degli assalti di eliminazione diretta		
4	fase successiva	0
3	fase successiva	0
5	2	7 3
67890	40000	2 4

Conferma



## UC11

Visualizzazione della classifica finale

Classifica finale					
posizione	codFIS	nome	cognome	CF	ranking
1	40000	Carlo	Bialilli	LERTNC45...	9.0
2	3	Lucia	Boro	LINPP5EAG...	2.0
3	4	Luisa	Ricci	LSABNC5E...	1.2
4	5	Carla	Privitera	CRIPPEA3...	0.5
5	2	Leandro	Boro	LBIBNC45E...	2.2
6	67890	Luigi	Bianchi	LBIBNC45A...	2.0

## UC10

Visualizzazione del nuovo ranking degli atleti

codFIS	nome	cognome	CF	genere	ranking
12345	Mario	Rossi	MRORSS4...	male	1.0
67890	Luigi	Bianchi	LBIBNC45A...	female	52.0
10000	Riccardo	Mascali	MROCCA5...	male	1.6
20000	Luigigigi	Bianchini	LBIABNAC4...	male	2.5
30000	Giuseppe	Signorini	MRRAS45A...	male	1.9
40000	Carlo	Bialilli	LERTNC45...	female	109.0
1	Luca	Rimo	MRORSS4...	male	7.0
2	Leandro	Boro	LBIBNC45E...	female	52.2
3	Lucia	Boro	LINPP5EAG...	female	82.0
4	Luisa	Ricci	LSABNC5E...	female	66.2
5	Carla	Privitera	CRIPPEA3...	female	65.5

## Sviluppi futuri

Esistono molteplici aspetti attraverso i quali è possibile ampliare il progetto. Il primo miglioramento da considerare riguarda l'introduzione di un database per la persistenza dei dati. Questo permetterebbe di salvare e recuperare le informazioni in modo efficiente e sicuro. Inoltre, potrebbe essere utile implementare la gestione di gare a squadre o di competizioni con format particolari, come ad esempio tornei con doppie tornate di gironi. Queste modifiche potrebbero arricchire notevolmente la complessità e la versatilità del progetto.