

# Programma G.I.A.

Tura Marco 0000658671

## Specifiche

Il presente documento si appresta a descrivere la fase progettuale e realizzativa del Programma G.I.A., applicazione finalizzato all'amministrazione delle gare indette da Apnea Academy Competition all'interno del circuito nazionale "Giro d'Italia in apnea".

Il presente documento si rifà al regolamento del suddetto circuito sportivo, versione 1.9 del Marzo 2017 ([link al regolamento](#)) e a diversi incontri con il committente che ne hanno specificato le fondamentali funzionalità e quelle meno necessarie, per tanto il programma non implementa tutti i funzionamenti che sono descritti nel regolamento ma solo un sottoinsieme di questi.

Di seguito si descriveranno in breve le funzionalità del programma.

Il programma in questione avrà il triplice scopo di:

- Gestire l'inserimento di tutti i dati degli atleti e delle società sportive partecipanti alle singole gare avendo cura di salvare e manipolare i dati al fine di semplificare la gestione amministrativa delle gare.
- Salvare i dati inerenti alle singole prestazioni dei partecipanti in ogni gara.
- Al termine delle singole gare manipolando i dati presi esportare la lista degli atleti, delle società ed i risultati della gara.

## Gestione delle società

Il programma deve permettere l'iscrizione, la cancellazione e l'eventuale modifica dei dati delle singole società che partecipano al circuito di gare, corredando ogni società dei dati necessari a renderla univoca all'interno dell'app.

- Nome della società
- Sede

Deve inoltre essere predisposto ad un'esportazione ed importazione dei dati delle società.

## Gestione degli atleti

Il programma deve permettere l'iscrizione, la cancellazione e l'eventuale modifica dei dati dei singoli atleti che partecipano al circuito di gare, corredando ogni atleta dei dati necessari a renderla univoca all'interno dell'applicazione.

- Nome
- Cognome
- Sesso
- Data di nascita
- Società di appartenenza
- Istruttore (SI/NO)
- Data di scadenza dell'idoneità sportiva

Vi sono da considerare vincoli funzionale sull'iscrizione degli atleti al sistema, ossia che l'età minima per cui è permessa la partecipazione è di 18 anni o 14 con l'autorizzazione di un genitore/tutore (all'atto dell'inserimento di un atleta con meno di 18 anni dovrà essere notificata la necessità di una autorizzazione).

Il sistema deve anche essere predisposto all'importazione ed esportazione dei dati degli atleti.

## Gestione gara

### Creazione

Il programma deve permettere la creazione di una gara associando ad essa le tipologie di discipline che la comporranno:

- Discipline Indoor [in piscina]
  - apnea statica (STA).
  - apnea dinamica con pinne (DYN).
  - apnea dinamica con monopinna (DYM).
  - apnea dinamica senza attrezzi (DNF).
  - apnea dinamica in immersione libera (FIO, Free Immersion Orizzontale).
- Discipline Outdoor [in acque libere]
  - assetto costante con pinne (CWF).
  - assetto costante con monopinna (CWM).
  - assetto costante senza attrezzi (CNF).
  - immersione libera (FIM).
  - campanelopetra (CAM).

deve inoltre dare la possibilità per ogni disciplina di aggiungere atleti partecipanti alla suddetta disciplina in modo non esclusivo, per cui un atleta che è iscritto alla gara può partecipare ad una sola disciplina o a tutte le discipline presenti in gara.

Inoltre per potersi iscrivere ad una gara si deve essere in possesso di un certificato medico valido (all'atto di inserimento di un atleta all'interno di una gara il sistema dovrà notificare all'utente la mancanza di un certificato medico valido se necessario).

### Amministrazione

Il sistema deve permettere ad un giudice di inserire le singole prestazioni di ogni partecipante alla gara avendo cura di discriminare tra le diverse discipline in cui un singolo atleta può partecipare all'interno della medesima gara, corredandole inoltre di ogni informazione necessaria a calcolare il punteggio: uso di attrezzature, penalità ed eventuali bonus.

I dati da inserire sono:

- Risultato della prestazione
  - Prestazione in secondi per la disciplina di apnea statica
  - Metri con arrotondamento al decimetro per le prove di apnea dinamica
  - Metri con arrotondamento al metro per prove di profondità
- Valutazioni tecniche (da 1 a 5 punti):
  - Assetto e postura

- Virata
  - Avanzamento
  - Bonus acquaticità
- Uso di attrezzature
  - Muta
  - Maschera
  - Tappanaso
  - Zavorra
- Malus percentili dovuti ad infrazioni (cartellino giallo).

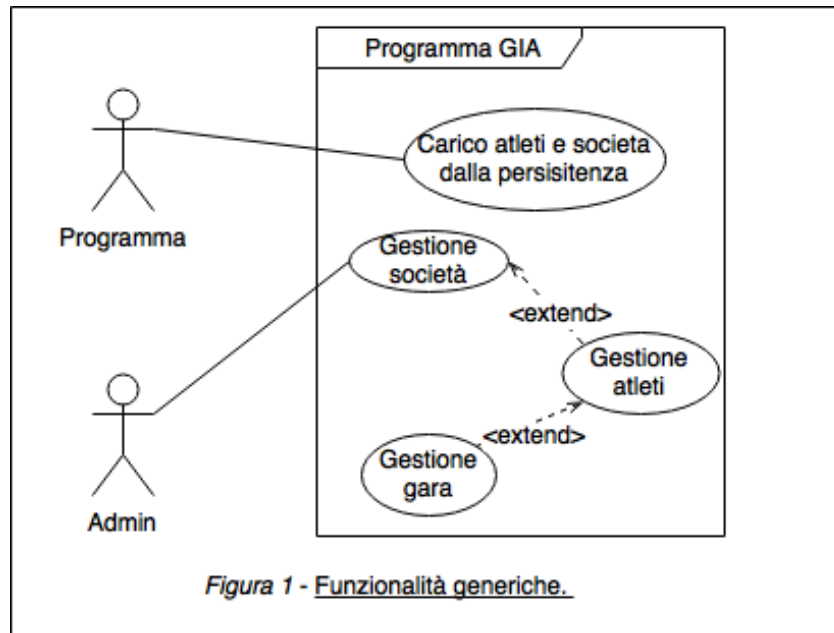
Sarà altresì possibile annullare la prova nel caso di grave infrazione da parte dell'atleta ponendo cartellini rossi.

Il sistema deve inoltre predisporre un automatismo per il calcolo dei punteggi delle singole prestazioni, diversificando il metodo di calcolo per ogni disciplina facente parte della gara.

Il sistema deve inoltre predisporre un metodo di esportazione delle informazioni riguardanti la gara gestita

*\*per maggiori informazioni sul calcolo dei punteggi e altre considerazioni vedi il [regolamento](#).*

## Casi d'uso



### GESTIONE GENERALE

**Attori:**

Admin  
Programma

**Precondizioni:**

Nessuna

**Post Condizioni:**

Nessuna

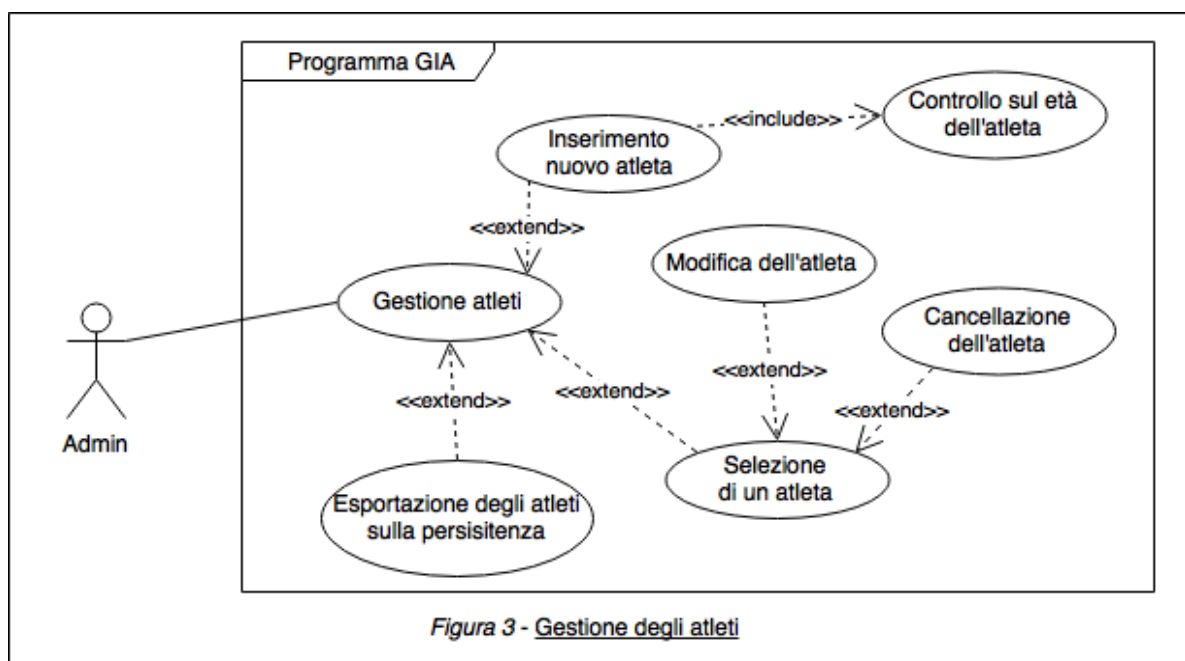
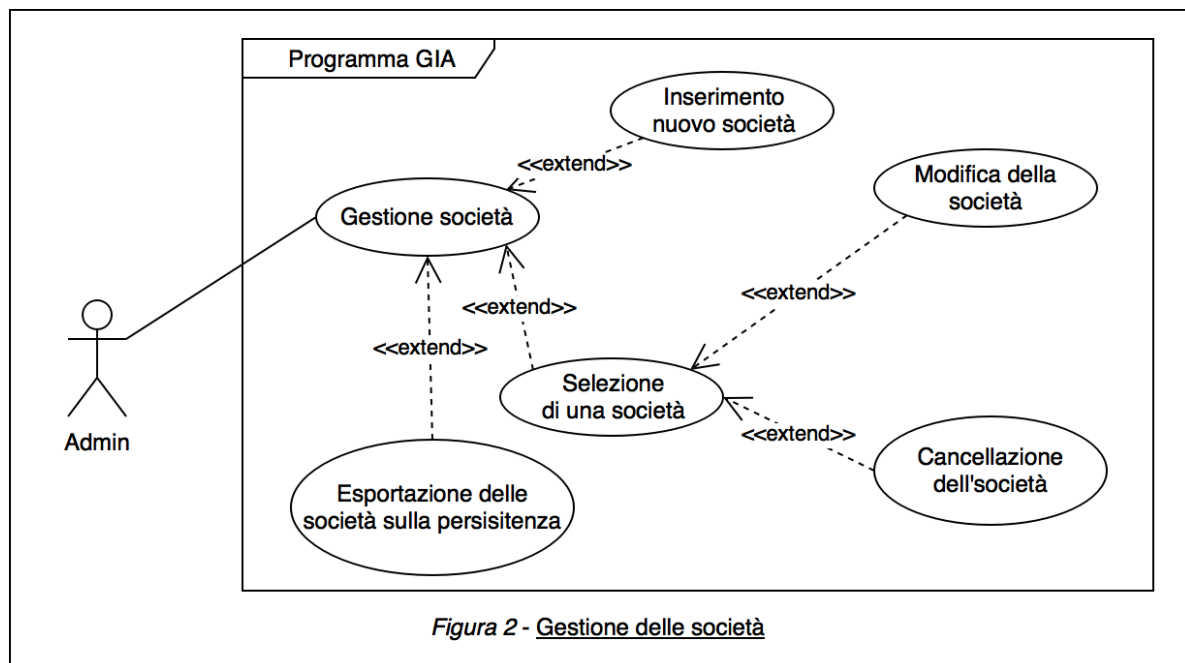
**Scenario Principale:**

1. Il programma carica società ed atleti dal dispositivo di persistenza.
2. L'admin inserisce, rimuove o edita le società
3. L'admin inserisce, rimuove o edita gli atleti e li associa alle società di appartenenza (non possono esserci atleti senza società).
4. L'admin, iscrive o disiscrive gli atleti alla gara.
5. L'admin gestisce la gara.

**Altro:**

L'inserimento postumo di società, utenti, o gare richiede di essere fatto seguendo la sequenza sopra indicata poiché ogni gara ha iscritti degli utenti ed ogni utente deve essere associato ad una società.

Nel caso non vi siano dati nel dispositivo di persistenza il sistema non caricherà nulla e salterà il passo 1



**INSERIMENTO DI UN NUOVO ATLETA****Attori:**

Admin

**Precondizioni:**

Esiste una società da associare al atleta.

**Post Condizioni:**

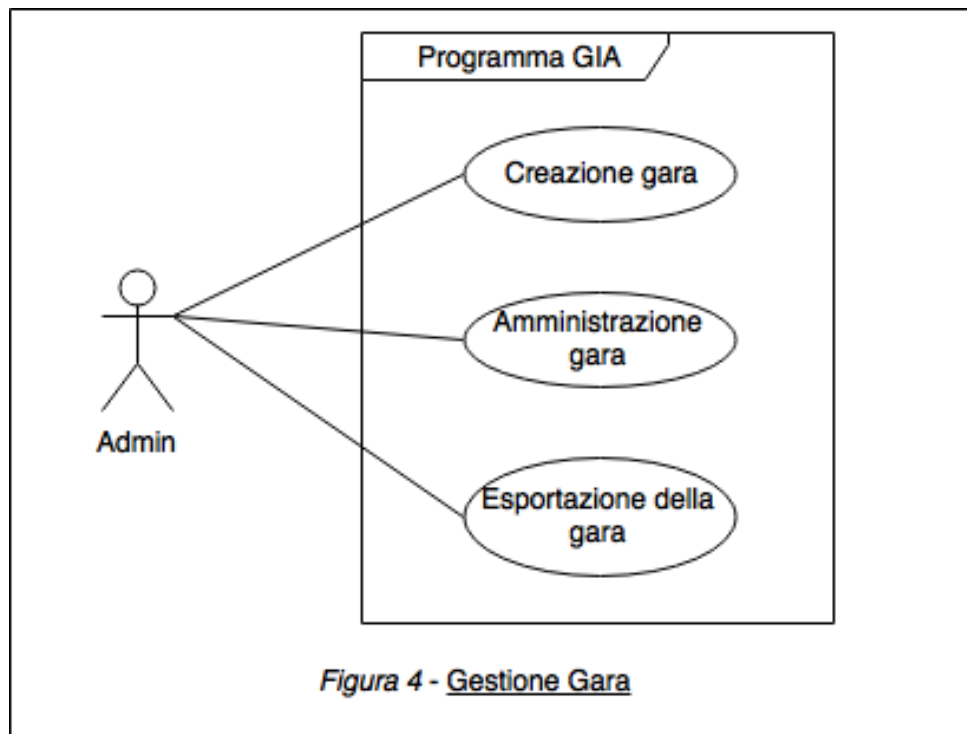
Esiste una nuovo atleta.

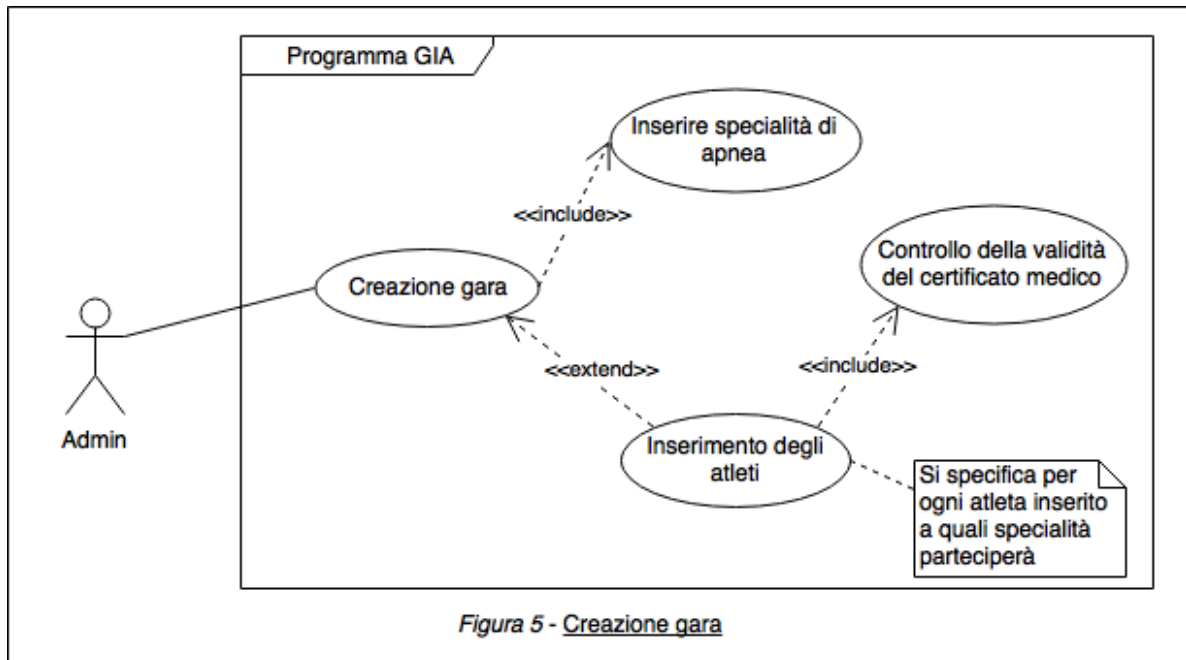
**Scenario Principale:**

1. L'admin seleziona una società da associare al nuovo atleta.
2. L'admin inserisce i dati associati al nuovo atleta.
3. il sistema controlla che l'età dell'atleta inserita sia inferiore a 18 anni e inferiore a 14 anni.
4. L'admin completa l'inserimento di un nuovo atleta.

**Scenario Alternativo:**

- 3a. Il sistema trova che l'età dell'atleta è inferiore di 18 anni ma superiore di 14 anni
  - 3a1. Il sistema apre una finestra informativa che ricorda la necessità di un autorizzazione.
  - 3a2. L'admin chiude la finestra informativa.
- 3b. Il sistema trova che l'età dell'atleta è inferiore di 14 o negativa.
  - 3a1. Il sistema apre una finestra informativa che avvisa l'impossibilità di aggiungere l'utente.
  - 3a2. Il sistema conclude l'inserimento dell'utente.





## CREAZIONE GARA

### Attori:

Admin

### Precondizioni:

Nessuna

### Post Condizioni:

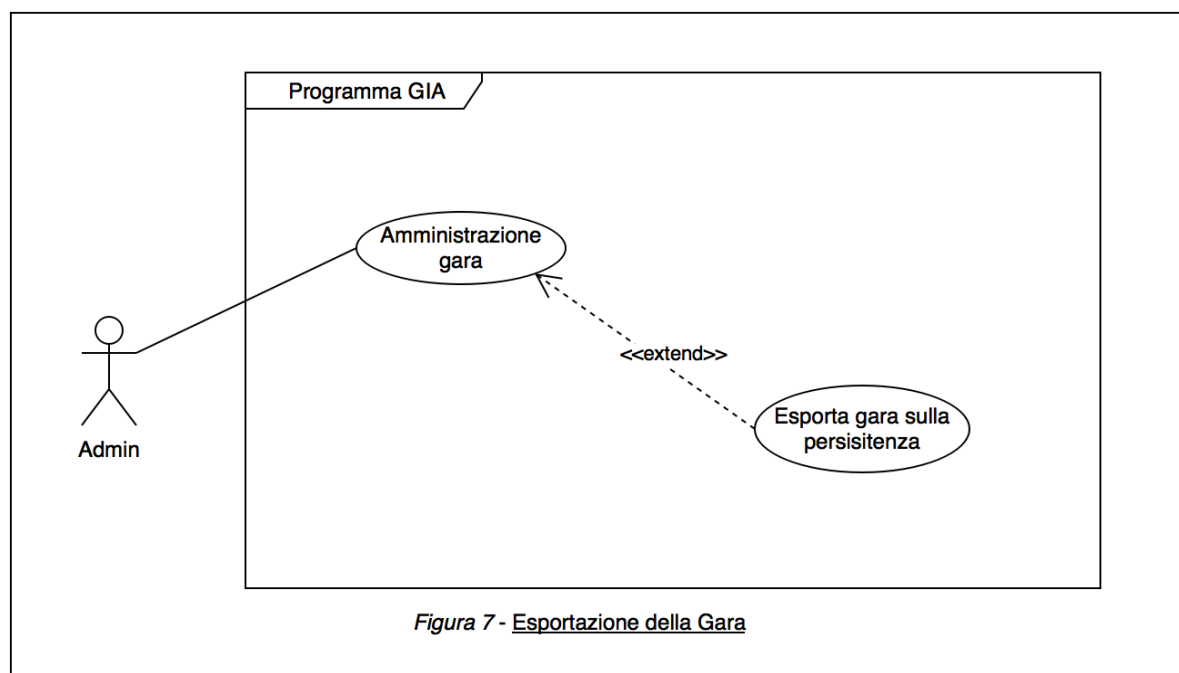
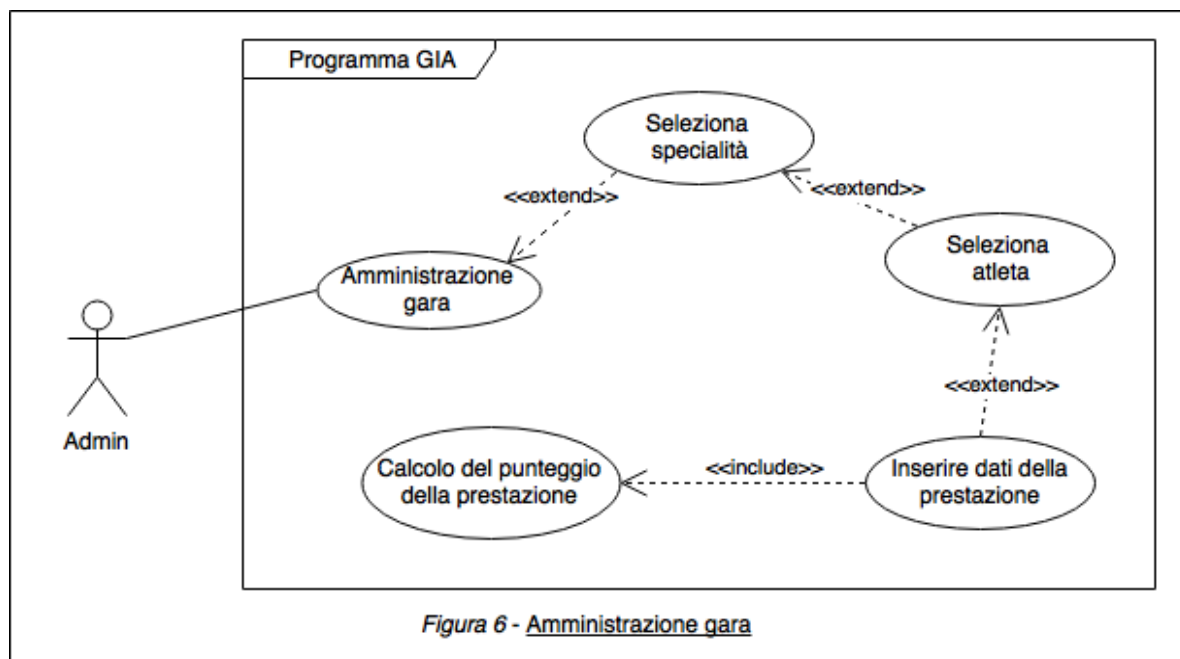
Esiste almeno 1 specialità nella gara

### Scenario Principale:

1. L'admin inserisce le specialità che saranno presenti in gara.
2. L'admin inserisce i partecipanti alla gara (passo ripetuto per ogni atleta che si vuole iscrivere).
  - a. L'admin seleziona un atleta da quelli presenti nel sistema
  - b. L'admin specifica per ogni atleta a quali specialità della gara parteciperà.
3. L'admin conclude la creazione di una nuova gara.

### Scenario Alternativo:

- 3a. L'admin cerca di inserire un atleta con il certificato scaduto
  - 3a1. Il sistema comunica all'utente l'impossibilità di inserire l'atleta.
  - 3a2. Il sistema ritorna al passo 2.

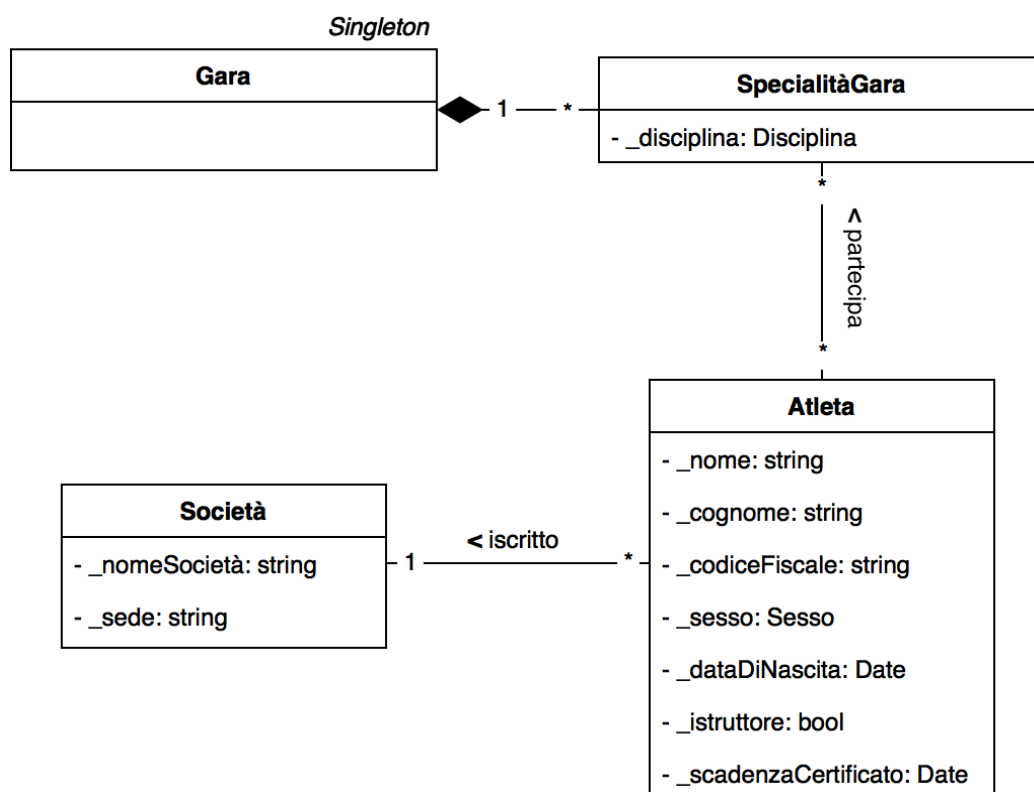




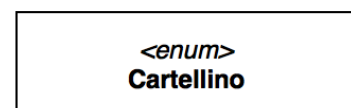
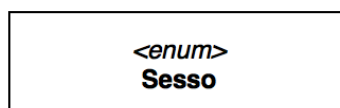
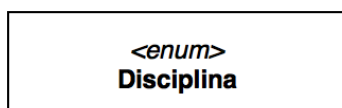
## UML di Analisi

Di seguito verranno illustrati gli UML di analisi, il primo di questi prende in considerazione le classi:

- Gara (Singleton)
  - 1+ specialità di gara
- Società
  - nome della società e sede
  - 0+ atleti iscritti
- Atleta
  - dati anagrafici vari, data di scadenza del certificato medico ed eventualmente può essere un istruttore (informazione questa utile solo al di fuori del dominio del programma).
  - 1 società di appartenenza per cui gareggia

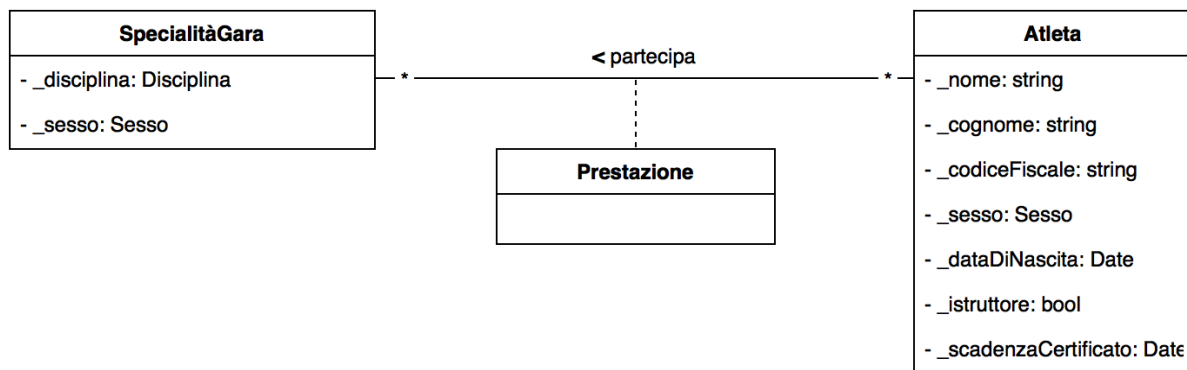


Sono inoltre presenti classi enumerative utili alla gestione del conteggio dei punteggi al fine della gestione delle classifiche.



Nel seguente diagramma delle classi UML viene approfondito il legame tra Atleti e specialità di gara:

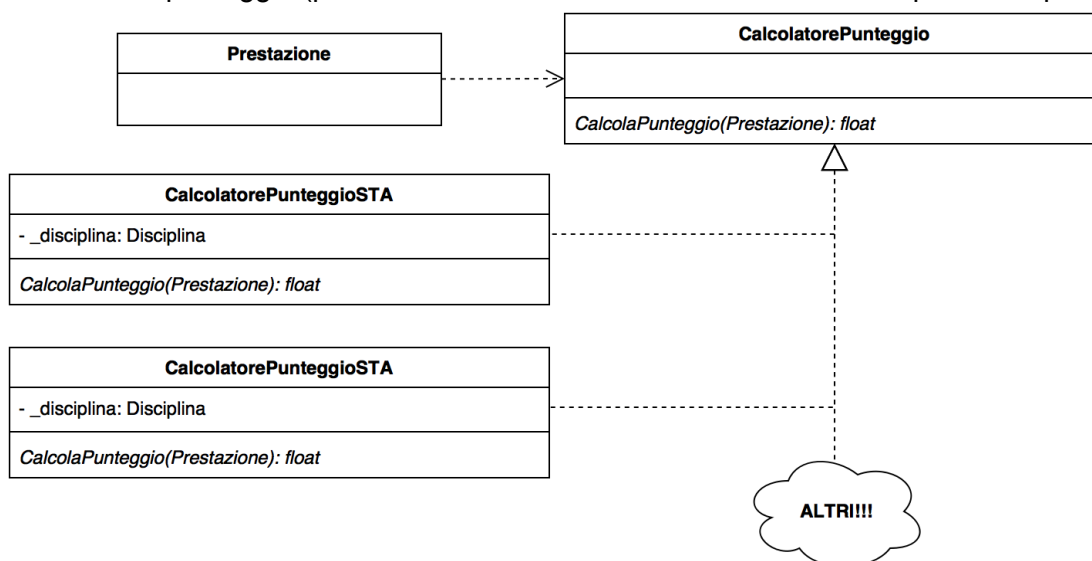
- SpecialitàGara
  - Disciplina specifica della specialita, scelta tra quella del enumerativo omonimo e sesso degli atleti che partecipano alla specialità anch'esso scelto dal enumerativo omonimo
  - 0+ atleti che partecipano alla specialità di gara con il vincolo di appartenere al sesso adeguato alla specialità
- Atleti
  - 0+ Specialità di gara a cui partecipano all'interno di una singola gara



Vi sarà inoltre una classe di associazione **Prestazione** che crea un legame 1 a 1 tra un **Atleta** ed una **SpecialitàGara** in modo che per ogni specialità di gara vi siano 0+ atleti che vi partecipino e ogni atleta abbia una prestazione riferita a quella specifica Specialità

- Prestazione
  - Punteggio della prestazione
  - Valutazioni tecniche (da 1 a 5 punti):
  - Malus percentili dovuti ad infrazioni (cartellini gialli, rossi).
  - Uso di attrezzature

Il punteggio delle prestazioni sarà calcolato attraverso un metodo specifico contenuto in una classe apposita chiamata **CalcolatorePunteggio**, questa oltre che ad implementare un adeguato algoritmo per ogni tipo di disciplina ingloba anche lo stato estrinseco necessario al calcolo del punteggio (parametro numerico variabile in funzione del tipo di disciplina ).

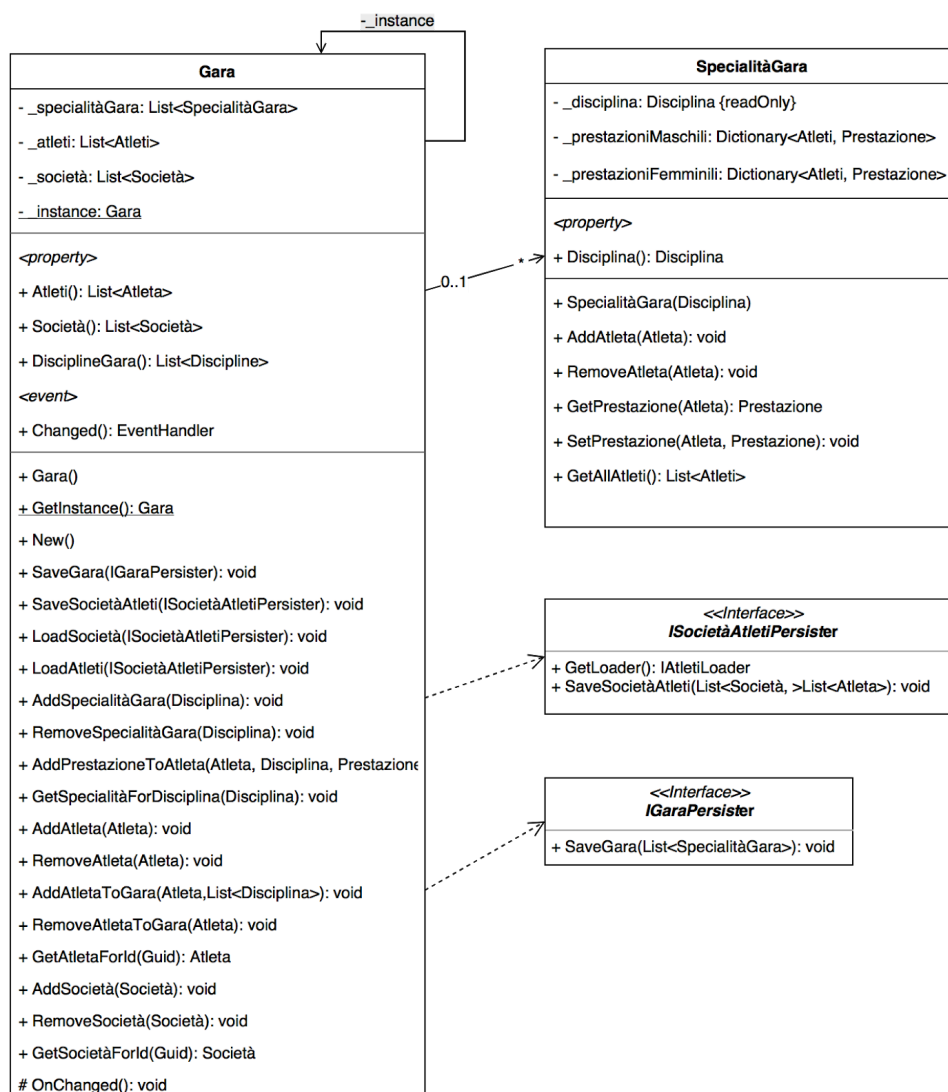


## UML di Progettazione

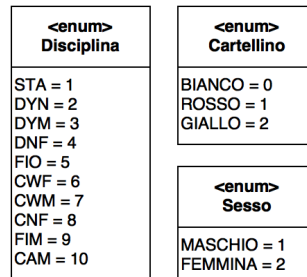
Di seguito verrà esposto il progetto del programma (attraverso UML) a partire da i diagrammi delle classi di analisi ed i casi d'uso visti sino ad ora.

Il programma sarà sviluppato in C# su framework .NET, sfruttando una persistenza realizzata su file XML ed una grafica di interazione basata su pattern MVP a view passiva. La classe Gara viene implementata attraverso un DP Singleton, in esso viene conservato lo stato di esecuzione del programma, caricando all'avvio la lista di atleti e società dal file XML attraverso le interfacce ISocietàAtletaPersister e viene mantenuto lo stato di avanzamento della gara sottoforma di lista di SpecialitàGara, da notare che la classe di associazione Prestazione è stata inglobata all'interno di un dizionario con chiave Atleta e valore Prestazione.

In fine la classe Gara attraverso l'interfaccia IGaraPersister da la possibilità di esportare i dati in un file XML dedicato, diverso da quello che contiene i dati degli atleti e delle società.

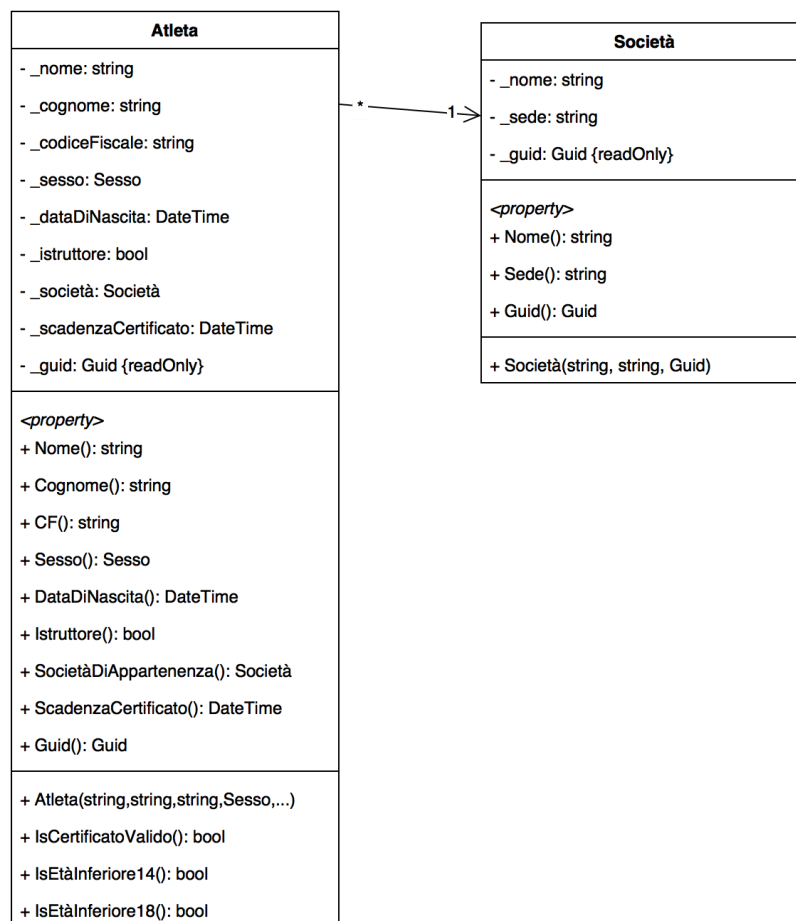


Al fine di gestire tutte le classi principali si è deciso di creare classi enumerabili per le discipline che compongono la gara, i cartellini che possono essere associati alle varie prestazioni ed al sesso dei partecipanti, quest'ultimo è un dato importante perché le varie specialità sono 2 per tipo, una maschile ed una femminile.



Ogni classe Atleta ha un'associazione con una Società, e viceversa ogni società è associata a più atleti.

Per esigenze grafiche il costruttore della classe ha solo 4 argomenti, in realtà gli argomenti sono 8, uno per ogni campo della classe, il campo “\_scadenzaCertificato” non è Read Only data la necessità di poterlo aggiornare, in caso contrario sarebbe necessario eliminare l'intero Atleta e reinserirlo completamente con il solo campo “\_scadenzaCertificato” alterato.

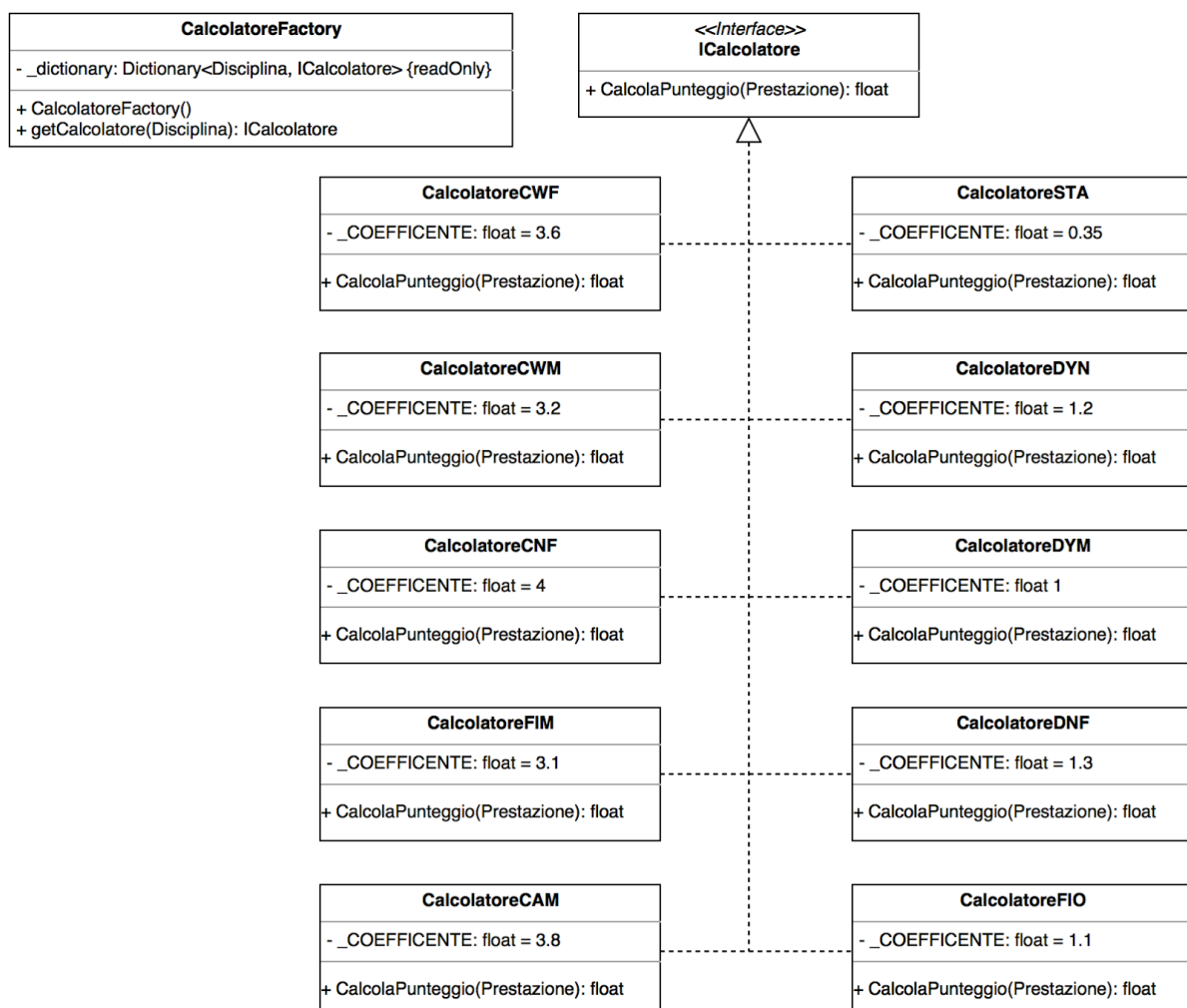


La classe Prestazione ha il compito di salvare a runtime tutte le informazioni necessarie al calcolo del punteggio della prestazione stessa, ogni disciplina che partecipa alla gara ha un suo specifico algoritmo di calcolo, ed un suo specifico parametro moltiplicatore, per tanto la funzione “CalcolaPunteggio()”, chiamerà a sua volta il metodo “getCalcolatore()” della classe CalcolatoreFactory passandogli la disciplina come parametro questa passerà la giusta istanza di Calcolatore contenente lo stato estrinseco

Prestazione
<ul style="list-style-type: none"><li>- _disciplina: Disciplina {readOnly}</li><li>- _misurazione: float</li><li>- _valutazioniTecniche_assetto: int</li><li>- _valutazioniTecniche_virata: int</li><li>- _valutazioniTecniche_avanzamento: int</li><li>- _valutazioniTecniche_acquaticità: int</li><li>- _attrezzatura_muta: bool</li><li>- _attrezzatura_maschera: bool</li><li>- _attrezzatura_tappanaso: bool</li><li>- _attrezzatura_zavorra: bool</li><li>- _cartellino: Cartellino</li><li>- _penalità: int</li><li>- _punteggio: int</li><li>- _prestazioneCompletata: bool</li></ul>
<p>&lt;property&gt;</p> <ul style="list-style-type: none"><li>+ Disciplina(): Disciplina</li><li>+ Misurazione(): float</li><li>+ VAssetto(): int</li><li>+ VVirata(): int</li><li>+ VAvanzamento(): int</li><li>+ VAcquaticità(): int</li><li>+ AMuta(): bool</li><li>+ AMaschera(): bool</li><li>+ ATappanaso(): bool</li><li>+ AZavorra(): bool</li><li>+ Cartellino(): Cartellino</li><li>+ Penalità(): int</li><li>+ IsCompletata(): bool</li></ul>
<ul style="list-style-type: none"><li>+ Prestazione(Disciplina, Sesso)</li><li>+ CalcolaPunteggio(Prestazione): float</li></ul>

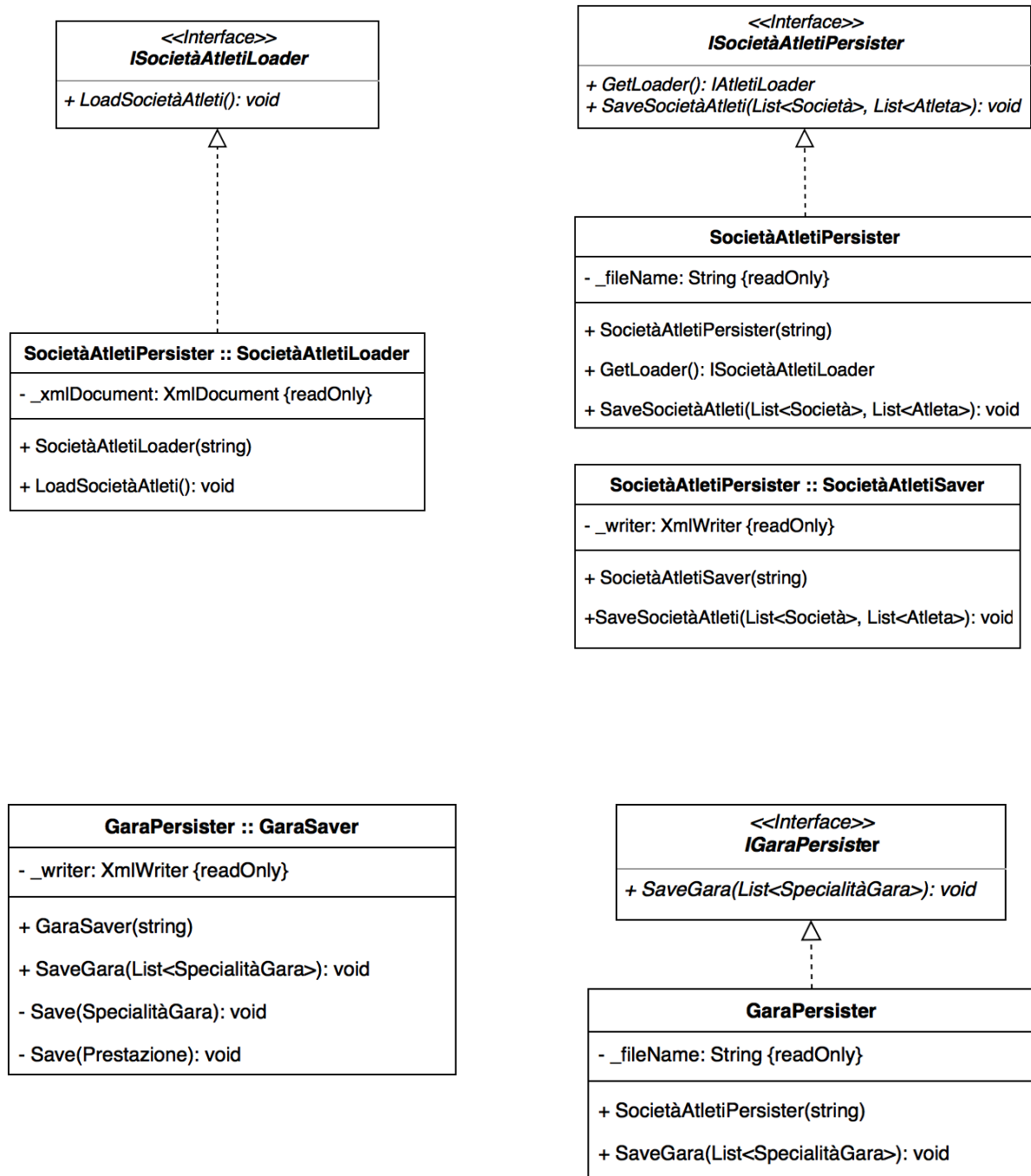
Il seguente UML espone un'ibridazione del pattern **Strategy** con il pattern **Flyweight**, in breve al fine di calcolare il corretto punteggio nelle singole prestazioni è necessario discriminare tra le varie discipline, si è quindi deciso di implementare i diversi algoritmi con classi concrete a partire da un'interfaccia omogenea tra i vari calcolatori, come nel pattern **Strategy**.

Si è inoltre deciso, come nel pattern **Flyweight**, di dividere lo stato intrinseco, contenente il coefficiente moltiplicativo specifico per la disciplina nella classe concreta CalcolatoreXXX e lo stato estrinseco, quindi dipendente dal contesto di funzionamento nella classe Prestazione, inoltre le istanze di CalcolatoreXXX vengono passate alla classe Prestazione, attraverso una classe Factory, così da garantire una corretta condivisione.

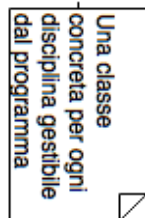


Infine gli ultimi diagrammi UML mostrano la progettazione delle componenti di caricamento e salvataggio delle classi Società, Atleta e Gara.

Da notare che mentre Società e Atleta fanno riferimento ad un singolo file XML, Gara viene esportato su di un file differente.



Di seguito viene mostrato un diagramma UML complessivo del progetto che tiene conto di tutte le classi di progettazione e i legami tra queste.





## Glossario

Termine	Significato	Sinonimi
Gara	Evento sportivo che il programma deve gestire	<ul style="list-style-type: none"><li>• Tappa</li></ul>
Atleta	Individuo che partecipa ad una gara	<ul style="list-style-type: none"><li>• Partecipante</li></ul>
Società	Società sportiva che partecipa al circuito di gare con una sua squadra di atleti	<ul style="list-style-type: none"><li>• Squadra</li></ul>
Prestazione	Risultato di un atleta all'interno di una gara per una disciplina specifica	
Punteggi	Risultato della prestazione di un atleta espressa in valore numerico	
Coefficiente di disciplina	Moltiplicatore numerico necessari per il calcolo del punteggio, varia da disciplina a disciplina	
Istruttore	Atleta con uno specifico brevetto che gli permette di accedere a determinate aree del campo gara (dato utile al di fuori del dominio dell'applicazione)	
Campo gara	Luogo dove si svolge la gara	
Idoneità sportiva	Idoneità medico sportiva che deve essere obbligatoriamente data dall'atleta all' organizzatore di gara per poter essere iscritto a quest'ultima	
Disciplina	Specifico tipologia di competizione, può essere indoor o outdoor	
Disciplina Indoor	Disciplina che si svolge	

	all'interno di una piscina	
Disciplina Outdoor	Disciplina che si svolge in mare aperto o in una piscina artificiale speciale	
Attrezzature	Attrezzature di ausilio alle prestazioni, possono essere: maschera, tappanaso, mute o collari zavorrati	
Cartellini	Penalità associata ad una prestazione, assegnati per gravi infrazioni al regolamento o per comportamenti pericolosi durante la prestazione. possono essere gialli o rossi	<ul style="list-style-type: none"><li>• penalità</li></ul>
Cartellino Giallo	Penalità percentuale al punteggio della prestazione, data da una infrazione delle regole durante la prestazione	
Cartellino Rosso	Penalità data da una infrazione grave delle regole durante la prestazione, causa l'annullamento della prestazione	
Cartellino Bianco	Cartellino associato ad una prestazione eseguita nel rispetto delle regole, non causa alcun genere di penalità	
Valutazioni tecniche	Valutazioni di specifici aspetti della prestazioni come: fluidità della pinneggiata e della virata e acquaticità dell'atleta	