



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Progetto Object Calciatori

MARIO PONTICIELLO

PASQUALE SOLE

N86004563

N86004581

Anno Accademico

23/24

Indice

1	Traccia progetto	3
1.1	Analisi Traccia	4
2	Modello Concettuale	6
2.1	Modello UML del dominio del problema	6
2.2	Modello UML del dominio della soluzione	7
2.2.1	Analisi UML del dominio della soluzione	8
3	Applicazione Java	8
4	Sequence Diagram	9
4.1	Funzionalità	9

1 Traccia progetto

Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI (Swing o JavaFX), per la gestione di calciatori di tutto il mondo.

Ogni calciatore è caratterizzato da nome, cognome, data di nascita, piede (sinistro, destro o ambidestro), uno o più ruoli di gioco (portiere, difensore, centrocampista, attaccante) e una serie di feature caratteristiche (ad esempio colpo di testa, tackle, rovesciata, etc.).

Il giocatore ha una carriera durante la quale può militare in diverse squadre di calcio. La militanza in una squadra è caratterizzata da uno o più periodi di tempo nei quali il giocatore era in quella squadra. Ogni periodo di tempo ha una data di inizio ed una data di fine. Durante la militanza del giocatore nella squadra si tiene conto del numero di partite giocate, del numero di goal segnati e del numero di goal subiti (applicabile solo ai giocatori di ruolo portiere). Il giocatore può inoltre vincere dei trofei, individuali o di squadra.

Il giocatore può avere anche una data di ritiro a seguito della quale decide di non giocare più. Le squadre di calcio sono specificate dal loro nome e nazionalità.

L'amministratore del sistema si identifica con una login ed una password e ha il diritto di inserire nuovi giocatori nella base di dati, modificarne i dati, aggiungere ulteriori informazioni oppure eliminare un giocatore.

L'utente generico può vedere l'elenco dei giocatori e le loro caratteristiche e può richiedere diverse ricerche, ad esempio filtrando i giocatori per nome, per ruolo, per piede, per numero di goal segnati, per numero di goal subiti, per età, per squadre di appartenenza.

1.1 Analisi Traccia

La traccia chiede di sviluppare un sistema informativo per la gestione di calciatori di tutto il mondo; quindi, nel nostro progetto teniamo in considerazione sia i calciatori di leghe alte, come serie A, Premier League ecc. e sia calciatori di leghe inferiori/giovanili, sia maschili che femminili. In questa analisi andremo a individuare tutte le informazioni presenti nel testo che definiranno la struttura del nostro database dei calciatori, in questa fase andremo a definire le varie entità e relazioni tra di loro, ed eventuali vincoli.

“Ogni calciatore è caratterizzato da nome, cognome, data di nascita, piede (sinistro, destro o ambidestro), uno o più ruoli di gioco (portiere, difensore, centrocampista, attaccante) e una serie di feature caratteristiche (ad esempio colpo di testa, tackle, rovesciata, etc.)”

Dal testo veniamo a conoscenza del fatto che i calciatori posseggono un Nome, Cognome, una data di nascita, qual è il loro piede preferito, abbiamo scelto di rappresentare queste informazioni come attributi dell'entità calciatore, mentre per quanto riguarda i ruoli creare un'entità a sé stante ci è sembrata la miglior opzione, andando a relazionare il giocatore con il ruolo, in questo modo un giocatore potrà ricoprire anche più ruoli diversi tra di loro. Stesso ragionamento per quanto riguarda le feature caratteristiche, abbiamo quindi creato una nuova entità che si andrà a relazionare con il calciatore permettendo a quest'ultimo di possedere zero, una o più feature, ovvero le loro migliori abilità come, ad esempio, il colpo di testa, rovesciata, pallonetto, punizioni dalla distanza ecc.

“Il giocatore può avere anche una data di ritiro a seguito della quale decide di non giocare più.”

In questo progetto si chiede anche di ricordare i giocatori che hanno smesso di giocare, tener traccia di tutte le loro informazioni aggiungendo una data che sta ad indicare il giorno in cui hanno smesso di giocare, abbiamo quindi optato di aggiungere una generalizzazione a calciatore chiamandola Calciatore ritirato, come unico attributo la data ritiro.

“Il giocatore ha una carriera durante la quale può militare in diverse squadre di calcio. La militanza in una squadra è caratterizzata da uno o più periodi di tempo nei quali il giocatore era in quella squadra. Ogni periodo di tempo ha una data di inizio ed una data di fine. Durante la militanza del giocatore nella squadra si tiene conto del numero di partite giocate, del numero di

goal segnati e del numero di goal subiti (applicabile solo ai giocatori di ruolo portiere).”

Sappiamo poi che i calciatori appartengono a una o più squadre, dobbiamo tenere traccia della loro data di inizio e fine militanza per ogni singola squadra in cui hanno giocato. Durante il periodo di permanenza per ogni squadra dobbiamo memorizzare tutti i gol segnati da quel calciatore e anche i gol subiti, statistica riservata solo ai giocatori che hanno ricoperto almeno una volta il ruolo di portiere. Tutte queste informazioni sono conservate nella classe associativa tra calciatore e squadra, tranne per i gol subiti, questa informazione deve apparire solo ai giocatori che hanno interpretato almeno una volta il ruolo di portiere, abbiamo quindi creato una generalizzazione di calciatore intitolata Portiere che non presenta attributi ma soltanto una classe associativa verso Squadra conservando i gol subiti e la data di inizio e fine militanza in quella squadra.

“Il giocatore può inoltre vincere dei trofei, individuali o di squadra”

Per soddisfare questa richiesta c'è bisogno di creare una nuova entità trofeo, andandola a differenziare tra quelli singoli e di squadra tramite un attributo “Tipo”, creiamo una classe associativa da trofeo a giocatore conservando anche la data di quando il trofeo è stato assegnato, inoltre visto che sono presenti anche i trofei di squadra, tutti e soli questi trofei verranno collegati con l'entità squadra tramite un'altra classe associativa conservando anche qui la data di assegnazione.

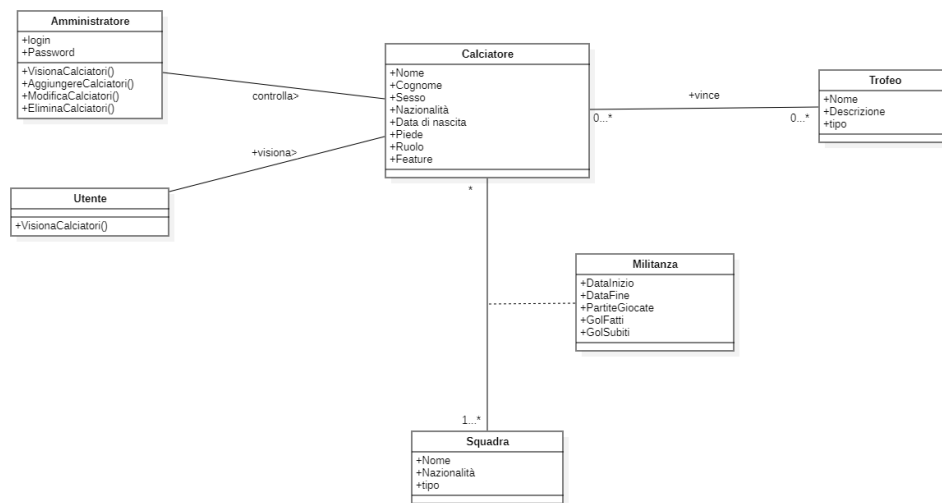
“Le squadre di calcio sono specificate dal loro nome e nazionalità.”

Abbiamo creato una nuova entità squadra che ha come attributi il suo nome, e la nazionalità da dove proviene, ogni squadra partecipa a diverse competizioni nazionali come il campionato o la coppa oppure europee/internazionali come la Champions League. Sia le squadre che le competizioni si dividono rispettivamente in sezione maschile e femminile tramite un attributo tipo.

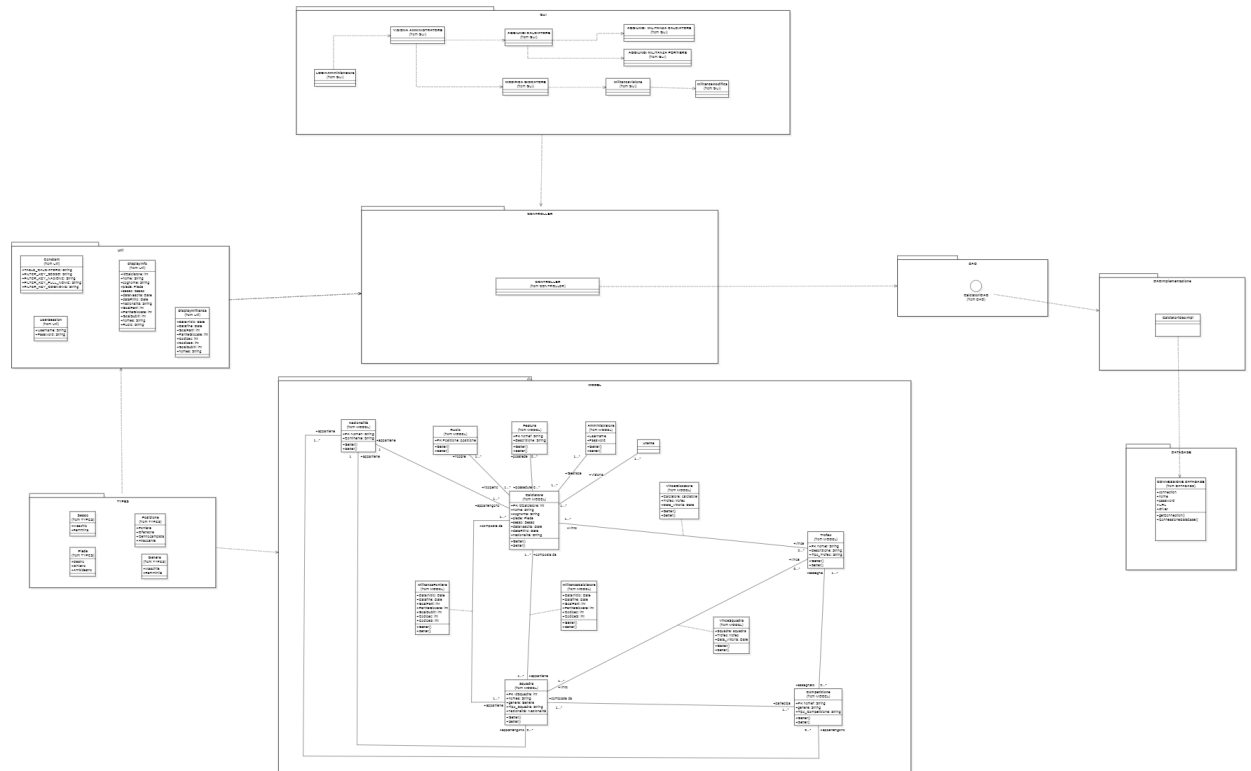
2 Modello Concettuale

2.1 Modello UML del dominio del problema

Modello UML costruito tramite le informazioni ottenute dall'analisi della traccia



2.2 Modello UML del dominio della soluzione



2.2.1 Analisi UML del dominio della soluzione

Abbiamo creato due classi **trofeo** differenziandoli per quei trofei che vengono assegnati al **singolo** e per quei trofei che vengono assegnati alle **squadre**, ovvero quelli che si vincono in una **competizione**.

Abbiamo separato da calciatore gli attributi **Nazionalità**, **Feature** e **Ruolo** e le abbiamo rese delle classi dato il loro legame con il calciatore che è di tipo **multi-a-molti**.

Abbiamo infine aggiunto la classe **competizione** che rappresenta appunto la competizione di cui una squadra può far parte in una determinata **stagione** e di cui può vincere il **trofeo assegnatogli**.

3 Applicazione Java

Ci troviamo nel nostro applicativo con 7 package:

- Model
- DAO
- Types
- DatabaseConnection
- Controller
- GUI
- Util

Nel **model** ritroviamo tutte le classi dell'UML con tutti i loro metodi **get**, **set** e **costruttore**.

Nel **DAO** ritroviamo l'interfaccia **CalciatoriDAO** che al suo interno possiede tutti i metodi poi usati nei vari controller, ed inoltre il metodo per la connessione al database.

Nel **database** ritroviamo invece il necessario per collegarci col nostro database **PostgreSQL**.

Nel **Controller** troviamo le classi controller che fanno da tramite tra il database e le gui prendendo e passando i dati richiesti.

In **Types** abbiamo gli attributi Enum **Sesso**, **Piede**, **Genere**, **Posizione**.

In **GUI** abbiamo invece tutte le gui del nostro applicativo.

In **Util** abbiamo le classi **Constant**,**DisplayInfo**,**DisplayMilitanza**,**UserSession** con all'interno metodi utili al tutto il nostro applicativo.

4 Sequence Diagram

4.1 Funzionalità

