



UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Documentazione progetto Object Orientation A.A 2023/2024

Sistema informativo dei calciatori

Autori: Orlowsky Gleb, Valentino Ferdinando, Scognamiglio Gianluca

Matricole: N86004789, N86004645, N86004467

Progettazione Orientata agli Oggetti del Sistema

Indice

1. Diagramma delle Classi del Dominio del Problema
 - 1.1 Class diagram
2. Comunicazione con il database
 - 2.1 Gestione della connessione al DB
3. Package DAO
 - 3.1 CareerDAO
 - 3.2 MatchDAO
 - 3.3 PlayerDAO
 - 3.4 TeamDAO
 - 3.5 UtilDAO
4. Graphical User Interface (GUI)
5. Sequence Diagram
6. Solution Diagram

1 Diagramma delle Classi del Dominio del Problema

Il diagramma delle classi del dominio del problema rappresenta le entità principali coinvolte nel sistema e le loro interazioni ad alto livello. Questo diagramma suddivide le classi in tre categorie: Boundary, Control ed Entity.

Il primo punto da sviluppare, prima di passare alla concreta realizzazione del progetto, è quella di analizzare la richiesta del nostro cliente. Nello specifico ci è stato richiesto di realizzare una base di dati relazionale, e di un rispettivo programma applicativo, che raccolga dati relativi a dei giocatori di calcio.

“Si sviluppi un sistema informativo 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.

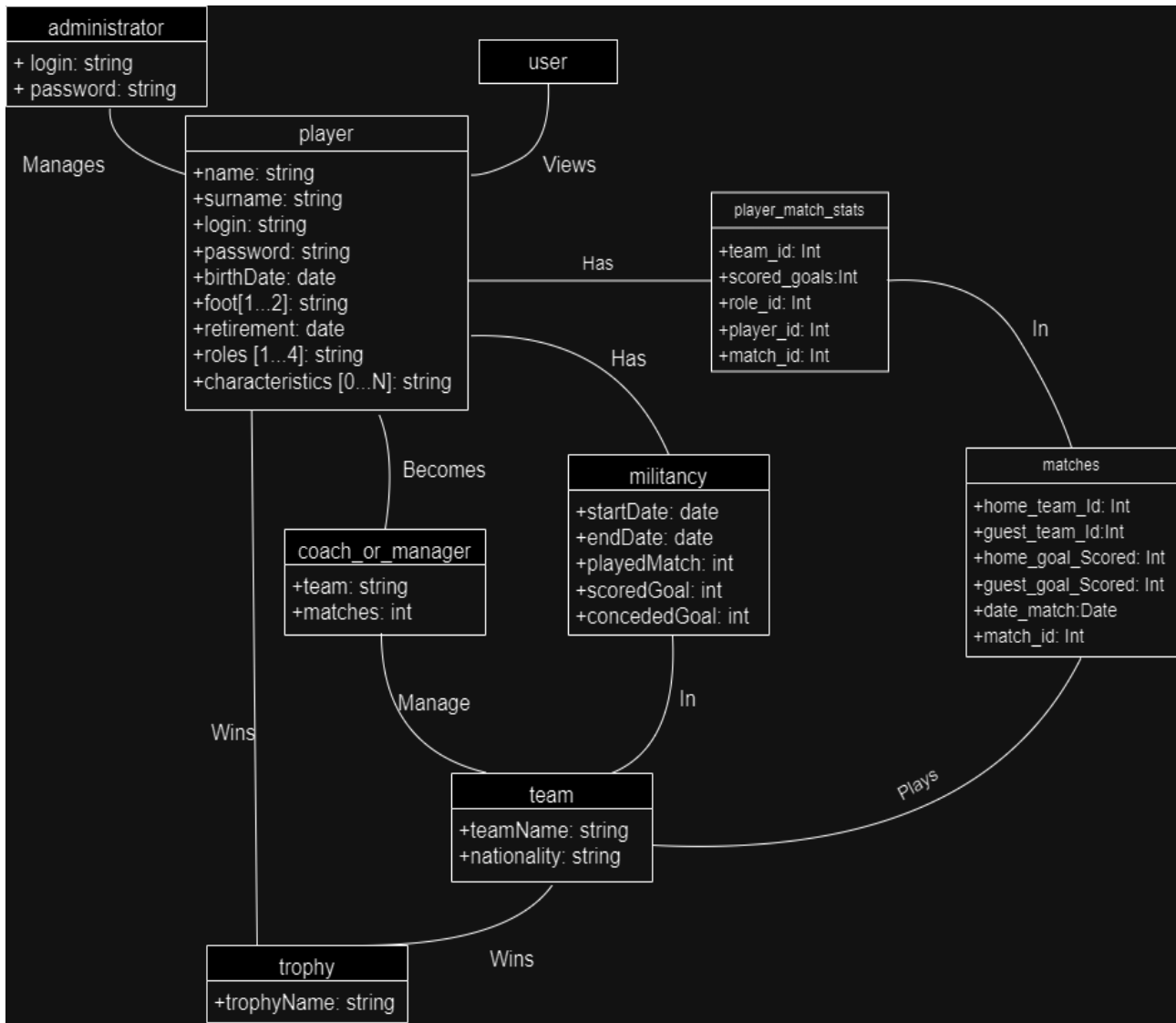
Il giocatore può avere anche una data di ritiro a seguito della quale decide di non giocare più. Le squadre di calcio sono specificate dal loro nome e nazionalità.

L'amministratore del sistema si identifica con un 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.

I giocatori dopo la fine della carriera possono diventare allenatori o dirigenti. Il sistema continua a mantenere una parte delle informazioni (squadra, numero di partite e trofei vinti) anche per allenatori e dirigenti.

Inoltre, può accedere al sistema anche un terzo tipo di utente, consistente nel Giocatore stesso. Egli ha un suo login e password e può modificare unicamente i dati relativi a sé stesso.

1.1 Class diagram



La classe del giocatore è forse quella più importante dove vertono tutte le altre.

Un giocatore viene associato ad una carriera, con una data d'inizio e una data di fine, la relazione tra il giocatore e la carriera è molti a una.

Il giocatore possiede un piede dominante o può essere ambidestro, il piede è posseduto da un giocatore, l'attributo piede è di tipo piede e verrà specificato proprio se usa il destro, il sinistro o entrambi.

Il giocatore potrà possedere anche diversi ruoli (portiere, difensore, attaccante e centrocampista), anche qui l'attributo ruoli è di tipo ruoli. Infine, potrebbe possedere anche diverse caratteristiche, oppure neanche una. Caratteristiche è un attributo di tipo caratteristiche.

La classe della carriera terrà in conto dei trofei che il giocatore ha vinto individualmente e dei trofei che ha vinto con le squadre in cui ha militato.

Durante la sua carriera il giocatore può militare diverse volte in diverse squadre di calcio, in periodi differenti. Nella classe della militanza, infatti, terremo conto del periodo di tempo in cui il giocatore è stato nella squadra.

Periodo è un attributo multivalore, il giocatore potrebbe anche cambiare squadra e ritornare nella vecchia successivamente, il giocatore può militare minimo 1 volta fino ad un numero “n” non definito di volte in una squadra (1: [1...*]), ma ad ogni periodo è associata una militanza.

La squadra è un attributo appartenente alla militanza. Nella classe squadra si terrà conto della nazionalità e del nome della stessa. Il giocatore può militare in una sola squadra per volta.

Coach e manager sono delle sottoclassi di giocatore ed ereditano alcuni dei suoi attributi. Stessa cosa vale per le classi giocatore e amministratore, sono delle sottoclassi di utente.

2 Comunicazione con il database

Per la comunicazione con il DB abbiamo utilizzato l'architettura DAO. Il pattern Data Access Object (DAO) è un pattern strutturale che ci permette di isolare il livello applicativo dal livello di persistenza utilizzando un'API astratta.

2.1 Gestione della connessione al DB

La connessione al DB è gestita dal pacchetto `db_connection` contenente la classe `DBConnection`, la quale si connette al DB utilizzando la libreria JDBC.

3 Package DAO

3.1 CareerDAO

Il pacchetto `CareerDAO` si occupa della gestione delle informazioni riguardanti la carriera di uno o più giocatori, come:

- Recupero delle informazioni della carriera di un giocatore
- La creazione di una nuova carriera
- Aggiornamento della data d'inizio di una carriera
- Aggiornamento della data di conclusione di una carriera
- Recupero dell'ID di una carriera
- Recupero dei nomi dei trofei

3.2 MatchDAO

Il pacchetto MatchDAO si occupa della gestione delle informazioni riguardanti le partite, come:

- Recupero della lista delle partite giocate durante il periodo di una determinata militanza
- Aggiunta di una nuova partita

3.3 PlayerDAO

Il pacchetto PlayerDAO si occupa della gestione delle informazioni riguardanti i singoli giocatori, come:

- Recupero delle credenziali durante il login di un giocatore
- Recupero delle informazioni dei giocatori di una determinata squadra
- Recupero delle informazioni di un giocatore con un determinato nome
- Aggiunta dei ruoli a un giocatore
- Aggiunta dei trofei a un giocatore
- Aggiunta di un nuovo giocatore
- Aggiunta del giocatore a una partita
- Recupero della lista completa dei giocatori
- Aggiunta della data di ritiro a un giocatore
- Eliminazione di un giocatore
- Recupero della data di ritiro di un giocatore
- Modifica dei dati di un giocatore
- Recupero della lista dei ruoli che occupa un giocatore
- Rimozione di un ruolo o una caratteristica di un giocatore
- Modifica delle caratteristiche di un giocatore

3.4 TeamDAO

Il pacchetto Team si occupa della gestione delle informazioni delle varie squadre di calcio

- Aggiunta di un trofeo alla squadra

- Recupero del nome di una squadra
- Recupero del nome dei trofei di squadra
- Recupero del ID di una squadra

3.5 UtilDAO

Il pacchetto utilDAO gestisce le informazioni degli utenti, in particolare degli admin, dei giocatori, dei coach e dei manager

- Recupero delle credenziali dell'utente

4. Graphical User Interface (GUI)

L'interfaccia grafica del nostro programma è stata realizzata con il framework Java Swing. L'interfaccia è gestita in seguente modo:

- Welcomeframe: Questa è la prima schermata che vede l'utente dopo aver avviato il programma, qui è possibile scegliere se entrare come un ospite o fare il login come un giocatore o un amministratore
- LoginFrame: Questa schermata compare se l'utente ha deciso di cliccare su login ed entrare come un giocatore o un amministratore. Qui è possibile inserire le proprie credenziali ed entrare cliccando su login oppure tornare alla schermata principale.
- MainFrame: Questa schermata contiene la lista di tutti i giocatori presenti nel database. In base a quale utente sta visionando il mainframe saranno disponibili diverse funzioni aggiuntive.

Nel caso dell'utente generico, tramite il mainFrame potrà visionare tutta la lista dei giocatori che fanno parte del DB, con le rispettive carriere. Potrà ordinare i giocatori per nome, cognome, età, data di ritiro, piede, ruolo, goal segnati o subiti e team. Inoltre, potrà ricercare i giocatori tramite il loro nome.

Nel caso di un utente di tipo "giocatore", oltre a tutte le funzionalità messe a disposizione per l'utente generico, potrà modificare le proprie informazioni personali, come il nome, il cognome o la propria password, e potrà anche ritirarsi definitivamente e porre fine alla sua propria carriera.

Nel caso invece di un utente di tipo admin, oltre a tutte le funzionalità messe a disposizione per l'utente generico e l'utente giocatore (fatta eccezione per il ritiro), potrà modificare ulteriormente le informazioni di un giocatore, come il ruolo, il piede, le caratteristiche i trofei e la sua carriera, potrà aggiungere nuovi giocatori e nuovi match, e potrà infine assegnare dei trofei di squadra.

- AddCareerFrame: Questa schermata viene utilizzata per aggiungere una nuova carriera ad un giocatore. È possibile inserire il team dove si svolgerà la nuova

carriera, la data di inizio e la data della fine della carriera. Questa schermata è resa disponibile solo agli utenti di tipo admin

- **AddMatchFrame:** Questa schermata viene utilizzata per aggiungere nuovi match. È possibile inserire la squadra che gioca in casa, la squadra avversaria e la data del match. Questa schermata è resa disponibile solo agli utenti di tipo admin
- **MatchesPlayedFrame:** Questa schermata permette di visualizzare tutti i match giocati da un giocatore durante una carriera, in particolare è possibile visualizzare i team che hanno preso parte al match, i goal di entrambe le squadre e la data dell'incontro. Questa schermata è resa disponibile a tutti gli utenti
- **PlayerAddFrame:** Questa schermata viene utilizzata per aggiungere dei nuovi giocatori. È possibile inserire un username, una password, un nome, un cognome, la data di nascita, il piede utilizzato dal giocatore e il ruolo del giocatore. Questa schermata è resa disponibile solo agli utenti di tipo admin
- **PlayerAddTrophyFrame:** Questa schermata ci permette di aggiungere trofei ai giocatori. È possibile inserire il tipo di trofeo vinto e la data di vincita del trofeo. Questa schermata è resa disponibile solo agli utenti di tipo admin.
- **PlayerModifyFrame:** Questa schermata permette di visionare e selezionare le informazioni dei giocatori che si desiderano cambiare.

Nel caso di un utente di tipo admin, ci permette di selezionare il nome, il cognome, la password, il piede, il ruolo, le caratteristiche, i trofei e le varie carriere.

Nel caso di un utente di tipo giocatore, ci permette di selezionare il nome, il cognome, la password e aggiunge l'opzione per ritirarsi e porre fine alla propria carriera da giocatore.

Questa schermata è resa disponibile solo agli utenti di tipo admin e giocatore.

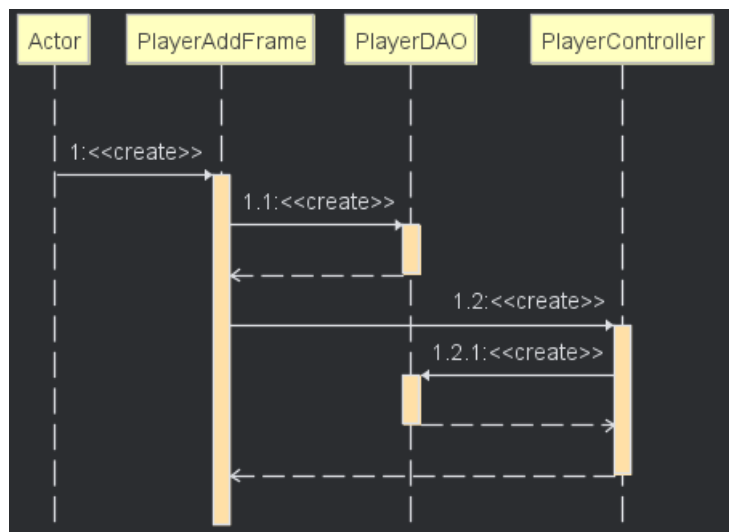
- **PlayerModifyAlterFrame:** Questa schermata è un'estensione di PlayerModifyFrame, e si occupa di modificare effettivamente le informazioni. Una volta selezionata una informazione su PlayerModifyFrame, apparirà questa schermata per cambiare l'informazione in questione. Questa schermata è accessibile solo agli utenti di tipo admin e giocatore.
- **PlayerProfileFrame:** Questa schermata ci permette di visualizzare le informazioni di un giocatore. Fornisce le informazioni come nome, cognome, data di nascita, piede utilizzato, caratteristiche, ruolo, i trofei che ha vinto e le sue carriere nelle varie squadre di calcio in cui ha giocato. Questa schermata è resa disponibile a tutti gli utenti.
- **SelectFormationFrame:** Questa schermata estende AddMatchFrame, e permette di selezionare tutti i giocatori che hanno giocato un match e i rispettivi goal segnati o subiti. Questa schermata è resa disponibile solo agli utenti di tipo admin.
- **CheckFrame:** Questa schermata appare ogni qual volta il programma chieda una conferma all'utente.

5 Sequence Diagram

I sequence diagram illustrano come le diverse parti del sistema interagiscono tra loro per eseguire specifiche funzionalità. Di seguito vengono mostrati i sequence diagram per l'aggiunta di un nuovo giocatore e per il recupero delle informazioni di carriera di un giocatore.

Aggiunta di un Nuovo Giocatore

Questo sequence diagram mostra come il PlayerController interagisce con il PlayerDAO e altre classi per aggiungere un nuovo giocatore al sistema.



Recupero delle Informazioni di Carriera di un Giocatore

```
sequenceDiagram
    actor Actor
    participant PlayerProfileFrame
    participant CareerController
    participant CareerDAO
    participant MatchController
    participant MatchDAO
    participant MatchesPlayedFrame

    Actor->>PlayerProfileFrame: 1-<<create>>
    activate PlayerProfileFrame
    PlayerProfileFrame->>CareerController: 1.1: convertCareerInfoToData
    activate CareerController
    CareerController->>CareerDAO: 1.1.1: formatDate
    activate CareerDAO
    CareerDAO-->>CareerController: 
    deactivate CareerDAO
    CareerController->>CareerDAO: 1.1.2: formatDate
    activate CareerDAO
    CareerDAO-->>CareerController: 
    deactivate CareerDAO
    CareerController->>CareerDAO: 1.2: formatDate
    activate CareerDAO
    CareerDAO-->>CareerController: 
    deactivate CareerDAO
    CareerController->>CareerDAO: 1.2.1: e ->
    activate CareerDAO
    CareerDAO->>CareerDAO: 1.2.1.1: parseDate
    activate CareerDAO
    CareerDAO-->>CareerController: 
    deactivate CareerDAO
    CareerController->>CareerDAO: 1.2.1.2: getCareerId
    activate CareerDAO
    CareerDAO->>CareerDAO: 1.2.1.2.1: getCareerId
    activate CareerDAO
    CareerDAO-->>CareerController: 
    deactivate CareerDAO
    CareerController->>MatchController: 1.2.1.3: getMatchesForCareerPeriod
    activate MatchController
    MatchController->>MatchDAO: 1.2.1.3.1: getMatchesForCareerPeriod
    activate MatchDAO
    MatchDAO-->>MatchController: 
    deactivate MatchDAO
    MatchController-->>PlayerProfileFrame: 
    deactivate MatchController
    PlayerProfileFrame->>MatchesPlayedFrame: 1.2.1.4: <<create>>
    activate MatchesPlayedFrame
    MatchesPlayedFrame-->>PlayerProfileFrame: 
    deactivate MatchesPlayedFrame
    PlayerProfileFrame-->>Actor: 
    deactivate PlayerProfileFrame
```

[illegible]