POLITECNICO DI TORINO

Dipartimento di Ingegneria Gestionale e della Produzione

**Corso di Laurea in Ingegneria Gestionale   
Classe L-8**

Tesi di Laurea

**Tool per le simulazioni di mercato di una rosa calcistica**

**Immagine che contiene orologio

Descrizione generata automaticamente**

**Relatore Candidato**

Prof. Corno Fulvio Maturo Alessandro Giovanni

AA. 2021/2022

**Sommario**

[1 Proposta di progetto 3](#_Toc111281026)

[1.1 Titolo della proposta 3](#_Toc111281027)

[1.2 Descrizione del problema proposto 3](#_Toc111281028)

[1.3 Descrizione della rilevanza gestionale del problema 3](#_Toc111281029)

[1.4 Descrizione dei data-set per la valutazione 3](#_Toc111281030)

[1.5 Descrizione preliminare degli algoritmi coinvolti 4](#_Toc111281031)

[1.6 Descrizione preliminare delle funzionalità previste per l’applicazione software 4](#_Toc111281032)

[1.7 Precisazioni 4](#_Toc111281033)

[2 Introduzione 5](#_Toc111281034)

[3 Descrizione del problema affrontato 5](#_Toc111281035)

[4 Descrizione del data-set utilizzato 5](#_Toc111281036)

[4.1 Struttura del database 5](#_Toc111281037)

[4.2 Tabella player 5](#_Toc111281038)

[4.3 Tabella playerdef 5](#_Toc111281039)

[5 Descrizione strutture dati e algoritmi utilizzati 6](#_Toc111281040)

[5.1 Descrizione strutture dati utilizzate 6](#_Toc111281041)

[5.2 Descrizione algoritmi utilizzati 6](#_Toc111281042)

[5.2.1 Algoritmo ricorsivo 6](#_Toc111281043)

[6 Diagramma delle classi principali 6](#_Toc111281044)

[7 Videate dell’applicazione e collegamento al video 6](#_Toc111281045)

[8 Tabelle con risultati sperimentali 6](#_Toc111281046)

[9 Conclusioni e valutazione dei risultati ottenuti 6](#_Toc111281047)

# 1 Proposta di progetto

## 1.1 Titolo della proposta

Tool per le simulazioni di mercato di una rosa calcistica.

## 1.2 Descrizione del problema proposto

Con il seguente software si vuole permettere all’utente finale di ottimizzare una rosa calcistica con determinate cessioni sul mercato valutando oppure imponendo vari parametri, alcuni economici come il valore di mercato del giocatore oppure il suo stipendio e altri legati alle performance con vari indicatori tratti da un data-set.  
Il problema è molto frequente all’interno dei club calcistici che devono effettuare operazioni di mercato in modo da trarre profitto sulle vendite e allo stesso tempo cercare sostituti evitando ridurre la forza della propria rosa.

## 1.3 Descrizione della rilevanza gestionale del problema

Questo tool sarebbe usato da un manager per provare a rinforzare la propria rosa negli indici in cui risulta carente. La grossa rilevanza gestionale del problema è che le valutazioni terranno conto delle risorse disponibili non simulando quindi operazioni di mercato irrealizzabili. Il software permetterà non solo di migliorare la propria rosa, ma permetterà anche, se voluto dall’utente, di limitare gli stipendi senza andare a ottenere un grosso decremento degli indici di performance legati alla squadra. Tutto ciò è possibile perché si potranno vedere come variano i parametri della propria squadra una volta che verranno prima simulate e poi nel caso effettuate le operazioni di mercato.

## 1.4 Descrizione dei data-set per la valutazione

Il data-set è stato tratto dal sito di Kaggle in cui sono presenti molti database che spaziano in diversi ambiti. Nel nostro caso, è tratto da un’applicazione chiamata Football Manager 2020 ed è presente al seguente link (<https://www.kaggle.com/datasets/ktyptorio/football-manager-2020?resource=download>) dove non vi è nessun vincolo di copyright.  
All’interno del nostro database oltre che ai valori economici, quali valutazione sul mercato e stipendio percepito, troviamo diversi indici legati alle performance dei singoli giocatori, di questi ne utilizzeremo solo alcuni se non il programma non sarebbe facilmente comprensibile da un utente non esperto nel settore.

## 1.5 Descrizione preliminare degli algoritmi coinvolti

Questo software sarà realizzato in linguaggio Java con l’ausilio delle interfacce grafiche in JavaFX, inoltre saranno usati i pattern MVC e il pattern DAO.  
Tra gli algoritmi coinvolti troviamo quelli basilari di ricerca di vari record che vengono effettuati tramite semplici interrogazioni SQL sul nostro database. Oltre a questi, troveremo algoritmi di ottimizzazione ricorsivi per trovare soluzioni ottimali che rispettano i vincoli imposti dall’utente. Inoltre, ci saranno anche algoritmi di simulazione, che con semplici operazioni matematiche, mostreranno come varierebbero i parametri dei diversi club con determinate operazioni di mercato. Per concludere saranno presenti diversi controlli per evitare problemi e aiutare l’utente nel caso si fosse dimenticato di inserire dei campi o effettuare operazioni.

## 1.6 Descrizione preliminare delle funzionalità previste per l’applicazione software

L’applicazione permetterà all’utente di immedesimarsi in un vero e proprio manager di una società calcistica. Dopo aver selezionato il club di cui si vorrà migliorare o semplicemente modificare la rosa (con un semplice utilizzo di una coppia di combo box), l’utente potrà simulare operazioni di mercato. A partire dalle varie informazioni sulla società che verranno mostrate, l’utente avrà due possibilità: la prima consiste nel ricercare alcuni giocatori che rispettano determinati parametri e osservare il caso di loro acquisto come varierebbero gli indicatori della rosa, la seconda invece prevederà la simulazione di una vendita di un proprio giocatore e, in base al valore ottenuto da questa cessione, l’algoritmo cercherà dei sostituti in modo da ottimizzare i parametri richiesti dal manager che eventualmente potrebbero non essere i punti di forza del giocatore di cui si cerca la cessione, ma parametri economici per abbassare per esempio lo stipendio medio della rosa.

## 1.7 Precisazioni

Sarà previsto un primo algoritmo di simulazione dove l’utente potrà scegliere degli indici di performance nei quali la squadra risulta carente, e trovare, con questo, delle soluzioni che rispettino un budget inserito per gli acquisti oltre che a un vincolo salariale, imposto anche questo dall'utente. L’algoritmo proporrà una soluzione per migliorare la rosa e, nell’eventuale ipotesi in cui non sia possibile, si occuperà di comunicare che con quei parametri non sono presenti combinazioni per migliorare gli indici della squadra.  
Inoltre, nel secondo algoritmo, quello principale, l’utente potrà selezionare dalla lista dei giocatori presenti in rosa un certo numero di calciatori appetibili per la vendita. Una volta selezionati e premuto un apposito bottone l’algoritmo calcolerà la migliore soluzione presente sul mercato che permetterà di ottimizzare gli indici della squadra non sforando il budget, in questo caso sarà il totale degli introiti ricavati dalla vendita (per semplicità si userà che il valore ricavato dalle cessioni è uguale alla somma dei valori attuali dei giocatori) e lo stipendio, ovvero la somma delle retribuzioni della possibile soluzione sarà minore o uguale della somma degli stipendi dei giocatori reputati ‘cedibili’ dall’utente.  
In questa parte sto valutando di inserire la possibilità di scelta se andare a cercare una soluzione che migliora la somma del valore complessivo degli indici oppure solo determinati indici selezionati proprio a partire dall’utente.  
Vorrei anche provare a inserire, una volta calcolate le possibili soluzioni degli algoritmi, dei diagrammi a barre per indicare come varierebbero gli indici delle prestazioni, il valore della rosa e il monte ingaggi delle soluzioni ottime proposte, in modo che l’utente possa visualizzare con più facilità i cambiamenti per vedere se effettivamente apportare le modifiche alla propria squadra.

# 

# 2 Introduzione

# 3 Descrizione del problema affrontato

L’applicazione realizzata permette di ottenere suggerimenti riguardanti le operazioni di mercato di una rosa calcistica.   
Un manager che tenta di migliorare la rosa della propria società ha due grossi problemi da risolvere: il primo riguarda l’acquisto di giocatori che dovranno migliorare determinati indici in cui la squadra risulta carente, il secondo invece riguarda la sostituzione dei propri calciatori con altri che permettono nuovamente di migliorare i punti di debolezza della squadra. Ovviamente in entrambi i punti, trattandosi di società calcistiche, ci saranno dei vincoli economici e tecnici da rispettare. Per quanto concerne il primo punto, infatti, il direttore sportivo dovrà avere un budget massimo per l’acquisto, un budget massimo per lo stipendio e un ruolo per cercare il futuro giocatore della squadra. Nel secondo punto, in genere, si cerca una soluzione ottima per rimpiazzare i calciatori con altri dello stesso ruolo il cui costo e la somma degli stipendi sia minore, ma allo stesso tempo permettano di migliorare o comunque non peggiorare gli indici della squadra.   
 Il programma tenta quindi di risolvere queste due criticità con l’ausilio di determinati algoritmi di ricerca e di ricorsione informatici, inoltre saranno anche presenti algoritmi di lettura e scrittura all’interno di un database contente tutti i calciatori tesserati per i migliori 5 campionati europei.

# 

4 Descrizione del data-set utilizzato

Il database utilizzato è stato tratto da un’applicazione chiamata “Football Manager” in cui l’utente si immedesima nel ruolo di un vero e proprio allenatore ma allo stesso tempo manager sportivo moderno provando a effettuare operazioni di mercato per migliorare la propria rosa oltre che ovviamente allenare la squadra.  
Il CSV del database è stato trovato sul sito Kaggle presso il seguente link: <https://www.kaggle.com/datasets/ktyptorio/football-manager-2020?resource=download> .

## 4.1 Struttura del database

Il database, inizialmente formato da una sola tabella chiamata player, successivamente è stato modificato per semplicità di utilizzo selezionando solo i giocatori che competono nei cinque migliori campionati Europei, ovvero: Premier League, La Liga, Serie A, Bundesliga e Ligue 1. Oltre a questa operazione di semplificazione il database è stato “pulito” in quando erano presenti errori di codifica dovuti alla conversione del database in un csv e non in un formato sql, il risultato finale è stato inserito all’interno della tabella palyerdef.



## 4.2 Tabella player

È la tabella originale, ha la stessa struttura della playerdef (illustrata successivamente), l’unica differenza è la presenza di errori di codifica e la presenza di tutti i campionati mondiali, non solo dei cinque migliori campionati europei.

Immagine che contiene tavolo

Descrizione generata automaticamente  
4.3 Tabella playerdef

Di seguito elenco solo gli attributi utilizzati all’interno del programma:

* number: identificativo univoco del giocatore (chiave primaria)
* name: nome del calciatore
* club: squadra di appartenenza del giocatore
* division: campionato in cui gioca il giocaotore
* based: dove è situato il campionato
* Immagine che contiene tavolo

  Descrizione generata automaticamente height: altezza del giocatore
* weight: peso del giocatore
* age: età del giocatore
* best\_pos: posizione/ruolo in cui il giocatore gioca meglio
* value: valore del giocatore wage: stipendio del giocatore
* tec: indice di tecnica del giocatore (da 0 a 20)
* str: indice di forza del giocatore (da 0 a 20)
* Immagine che contiene tavolo

  Descrizione generata automaticamente pas: indice di passaggio del giocatore (da 0 a 20)
* mar: indice di marcamento del giocatore (da 0 a 20)
* pos: indice di posizionamento dei giocatori

# 

# 5 Descrizione strutture dati e algoritmi utilizzati

## 5.1 Descrizione strutture dati utilizzate

L’applicazione è stata realizzata utilizzando il linguaggio di programmazione Java per la gestione della logica applicativa e JavaFX per gestire la componente grafica dell’applicazione.  
Il tutto è stato realizzato mediante il pattern MVC (Model-View-Controller) che permette di separare all’interno del software la logica applicativa, l’interfaccia grafica e la struttura dati.  
Infatti, l’applicazione è suddivisa in tre package distinti:

* Il package *it.polito.tdp.FootballManager* contiene tutte le classi dedicate alla realizzazione della parte grafica dell’applicazione, dove l’utente inserirà i vari parametri di input e visualizzerà i risultati ottenuti.

Immagine che contiene testo

Descrizione generata automaticamente

* Il package *it.polito.tdp.FootballManager.db* si comporta come DAO (Data Access Object) gestendo quindi l’interazione e l’estrazione dei dati dal database per il corretto funzionamento del software.

Immagine che contiene testo

Descrizione generata automaticamente

* Il package *it.polito.tdp.FootballManager.model* si comporta da model quindi contiene tutte le classi legate alla componente algoritmica che gestiscono l’elaborazione dei dati.

Immagine che contiene testo

Descrizione generata automaticamente

## 5.2 Descrizione algoritmi utilizzati

Per realizzare il programma sono stati utilizzati molti algoritmi, tra i più importanti ricordiamo sicuramente i vari algoritmi di lettura da database, i vari algoritmi usati per semplificare e correggere il database, i diversi algoritmi utilizzati dai metodi all’interno delle classi e come più importante l’algoritmo ricorsivo.

### 5.2.1 Algoritmo ricorsivo

*In*[*informatica*](https://it.wikipedia.org/wiki/Informatica)*viene detto****algoritmo ricorsivo****un*[*algoritmo*](https://it.wikipedia.org/wiki/Algoritmo)*espresso in termini di se stesso, ovvero in cui l'esecuzione dell'algoritmo su un insieme di dati comporta la semplificazione o suddivisione dell'insieme di dati e l'applicazione dello stesso algoritmo agli insiemi di dati semplificati.   
(*Fonte Wikipedia)

All’interno di questo tool l’algoritmo è stato utilizzato nella sezione di mercato. Essendo un algoritmo molto complesso che richiama se stesso più volte al su interno sono state adottati alcuni filtraggi per permettere un corretto funzionamento. Nella sezione di mercato è possibile, infatti, scegliere un certo numero di giocatori destinati alla vendita, a questo punto per ciascun giocatore verranno selezionati i migliori 10 giocatori nel suo ruolo che migliorano la media degli indici della squadra e rispettano alcuni vincoli. I due vincoli sono che il valore del giocatore “possibile” e allo stesso tempo il suo stipendio devono essere inferiori o uguali al giocatore che è stato destinato alla vendita. Questa operazione verrà ripetuta più volte per ciascun giocatore destinato alla vendita andando a creare una lista di giocatori possibili sulla quale andrà a ciclare l’algoritmo ricorsivo. L’algoritmo ricorsivo andrà a controllare la soluzione ogni volta che la soluzione parziale avrà dimensione uguale al numero di giocatori destinati alla vendita. Oltre a questo controllo vi sarà un controllo sui ruoli, ovvero il numero di giocatori per ciascun ruolo destinati alla vendita dovrà essere uguale a quello della soluzione. Infine, la soluzione non verrà scartata solo se il totale degli indici di questa soluzione parziale sarà maggiore o uguale (da controllare) della soluzione ottima.

METTI SCHERMATE E SPIEGA

### 

# 6 Diagramma delle classi principali

# 7 Videate dell’applicazione e collegamento al video

# 8 Tabelle con risultati sperimentali

# 9 Conclusioni e valutazione dei risultati ottenuti