**Relazione finale**

**PROGRAMMA GESTIONALE DEL FANTACALCIO**

Per realizzare il *Programma Gestionale del Fantacalcio*, si è proceduto a compiere diverse fasi di analisi, descritte essenzialmente dalla stesura di tre diversi documenti:

* un’**analisi dei requisiti**, in cui sono stati studiati i requisiti che avrebbe richiesto il software;
* un’**analisi funzionale**, che ha consentito di stabilire le funzioni del programma;
* un’**analisi tecnica**, in cui è stata data una prima descrizione di tutti gli aspetti tecnici legati al software.

Successivamente a questi studi, si è proceduto alla programmazione vera e propria, che ha portato diversi cambiamenti rispetto a quanto preventivato inizialmente: è per questo motivo che si rende necessario, in questa relazione finale, un riepilogo delle analisi iniziali con le nuove caratteristiche, in modo da comprendere correttamente ogni aspetto legato al programma.

**Tempi**

Per l’esecuzione del progetto sono stati definiti dei tempi precisi di consegna:

* Consegna delle analisi dei requisiti, funzionale e tecnica entro il 22/10/2021;
* Consegna del codice sorgente del programma e del file eseguibile entro il 12/11/2021.

**Analisi dei requisiti**

**La situazione iniziale**

Viene richiesta la creazione di un programma per computer desktop che permetta la gestione dei tornei di Fantacalcio.

**Scopo e funzionamento del gioco**

Il **Fantacalcio** è un gioco che simula partite e tornei utilizzando le gesta dei veri giocatori del campionato. A questo gioco possono giocare un minimo di due giocatori (che vengono chiamati “**fanta-allenatori**”), fino ad arrivare ad un massimo di 10.  
Lo scopo è quello di formare una “**fanta-squadra**”, cioè una squadra immaginaria, formata dai giocatori di un torneo reale, come la **Serie A**, la **Champions League** oppure la **Premier League inglese**, per poi accumulare punteggi in base alle prestazioni dei calciatori che sono stati scelti.  
Il primo momento che sancisce l'inizio del **Fantacalcio** è l’asta, durante la quale i partecipanti del gioco formano le proprie squadre. L’asta prevede diversi fondamentali passaggi:

1. Prima di tutto è necessaria una lista contenente l'elenco di tutti i giocatori del campionato di riferimento. In queste tabelle ad ogni calciatore è associato un nome, un cognome, una squadra di appartenenza, un numero di maglia, un ruolo (che può essere attaccante, difensore, portiere e centrocampista) e un numero, detto quotazione iniziale, che ne rappresenta il valore "**fantacalcistico**".
2. Ogni **fanta-allenatore** parte con un budget stabilito e uguale per tutti (per esempio 260 Fantamilioni). Tale somma serve per acquistare i giocatori e formare la squadra. Un giocatore può giocare in una sola squadra.
3. Viene dato inizio alla fase più importante dell’asta: i partecipanti si sfidano a colpi di offerte, che non hanno limiti nei rilanci, per aggiudicarsi i giocatori e chi offre di più per un determinato calciatore, se lo aggiudica in esclusiva.
4. Alla fine dell'asta ogni partecipante deve avere a disposizione una “**rosa**” di **25 giocatori**: **3 portieri**, **8 difensori**, **8 centrocampisti** e **6 attaccanti**.

Completate le squadre, inizia il gioco, che durerà per tutta la stagione calcistica.  
Ogni giornata di campionato reale, ovvero quello giocato negli stadi, i **fanta-allenatori** schierano una squadra composta da 11 calciatori scelti dalle **rose** a loro disposizione, più le **riserve** (al massimo sette) che potranno sostituire eventuali calciatori che non scenderanno in campo o che per i motivi più disparati non prenderanno voto. Le prestazioni dei calciatori scelti determineranno l'andamento della squadra.  
I punteggi dei giocatori sono dati dai voti delle pagelle individuali che siti e quotidiani sportivi stilano al termine delle partite. A questi voti poi si aggiungono dei **bonus** o dei **malus** che aumentano o diminuiscono il punteggio totale della prestazione del calciatore:

* +3 punti per ogni **gol** segnato;
* +3 punti per ogni **rigore parato** (portiere);
* +2 punti per ogni **rigore segnato**;
* +1 punto per ogni **assist** effettuato;
* -0,5 punti per ogni **ammonizione**;
* +1 **portiere** **imbattuto**;
* -1 punto per ogni **gol** **subito** dal portiere;
* -1 punto per ogni **espulsione**;
* -2 punti per ogni **autorete**;
* -3 punti per un **rigore sbagliato**.

Alla fine del campionato reale, vince chi avrà accumulato più punti nella classifica generale.

**Scopo del programma**

Lo scopo del programma non è quello di fornire ai fanta-allenatori il gioco vero e proprio del Fantacalcio, bensì di permettere la registrazione dei dati di gioco, assumendo la forma di programma gestionale: molte delle parti del gioco, come l’asta, devono essere gestite autonomamente dai fanta-allenatori e poi inserite nel programma, cercando quindi di migliorare notevolmente la gestione dei dati per loro.

**Chi deve usare il programma?**

Il programma può essere utilizzato da un solo utente di qualunque tipo che abbia compreso appieno il funzionamento del gioco, illustrati all’interno dell’analisi funzionale.

**Requisiti software**

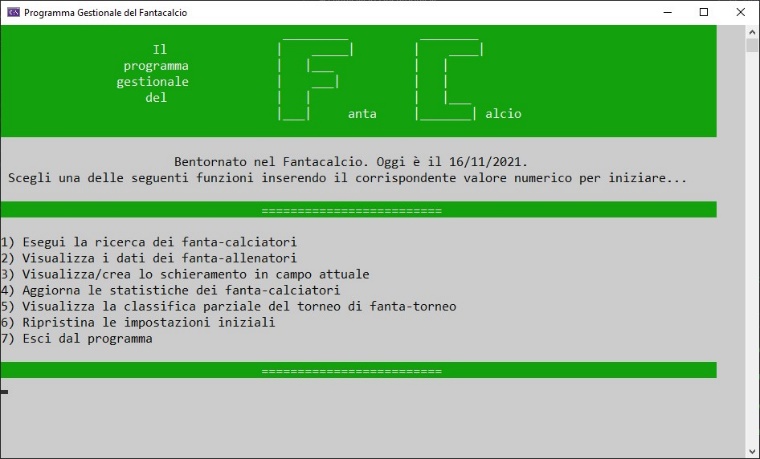
Il programma è un’applicazione basata su .NET Core 3.1, compatibile con Windows, MacOS e Linux.

**Requisiti hardware necessari**

Il programma, essendo estremamente semplice, necessita di un PC e di una tastiera per l’inserimento dei comandi da tastiera.

**Analisi funzionale**

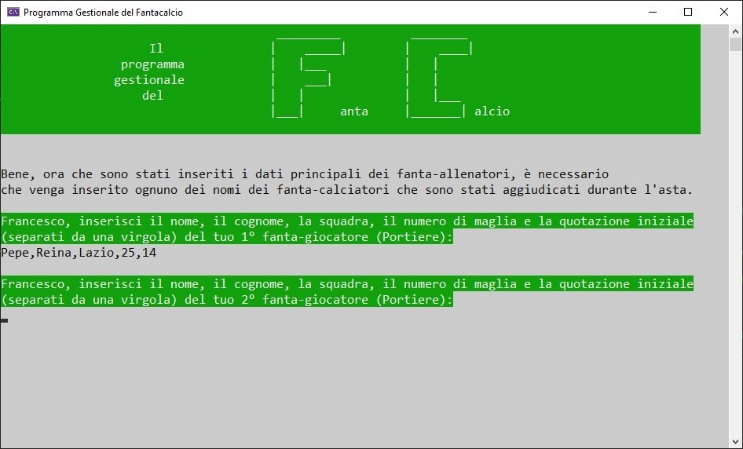
Per realizzare il programma gestionale del Fantacalcio non si utilizzerà una GUI (Graphical User Interface), ma un’interfaccia di tipo **CLI** (Command Line Interface, di tipo console).   
Di seguito verrà spiegato dettagliatamente il funzionamento del programma.   
Successivamente, quando si dirà “il programma richiede l’inserimento”, si intende che l’utente debba inserire un comando da tastiera, composto da una o più lettere, e poi deve premere invio per confermare l’inserimento.

**Schermata iniziale**

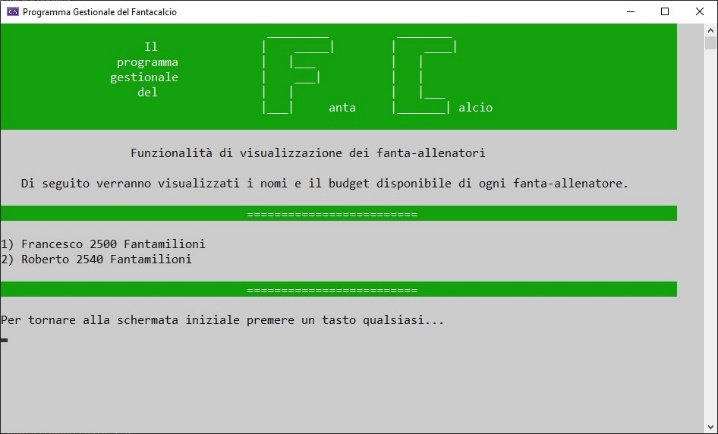
All’avvio del software viene presentata una schermata iniziale, nella quale vengono fornite all’utente alcune indicazioni di base su come funziona il programma. In base al momento in cui ci si trova e alle richieste del software in questa schermata è possibile distinguere due situazioni differenti.

* Se è il **primo avvio**, cioè nessun campionato di Fantacalcio è stato ancora impostato, il programma richiede all’utente l’inserimento, innanzitutto, del torneo calcistico reale scelto dai partecipanti. Successivamente chiede l’inserimento del numero di fanta-allenatori che partecipano al gioco, oltre al nome e al budget disponibile di ciascuno di loro. Una volta compiuto quest’ultimo passaggio, la console viene pulita e si passa alla schermata di inserimento delle rose dei giocatori.
* Se è un **avvio comune**, invece, in cui sono già stati configurati i dati di gioco, viene visualizzato un menu (come quello dell’immagine di fianco) di scelta tra un elenco numerato di opzioni:
  + **Ricercare** uno o più fanta-calciatori, insieme alle loro caratteristiche (inserendo il comando **1**);
  + **Visualizzare i dati** dei fanta-allenatori (con il comando **2**)
  + **Visualizzare** oppure **effettuare** **lo schieramento** in campo dei giocatori (con il comando **3**);
  + **Aggiornare** le statistiche dei giocatori (con il comando **4**);
  + **Visualizzare la classifica** parziale (con il comando **5**);
  + **Cancellare** i dati del torneo di Fantacalcio (con il comando **6**);
  + **Uscire** dal programma (con il comando **7**).

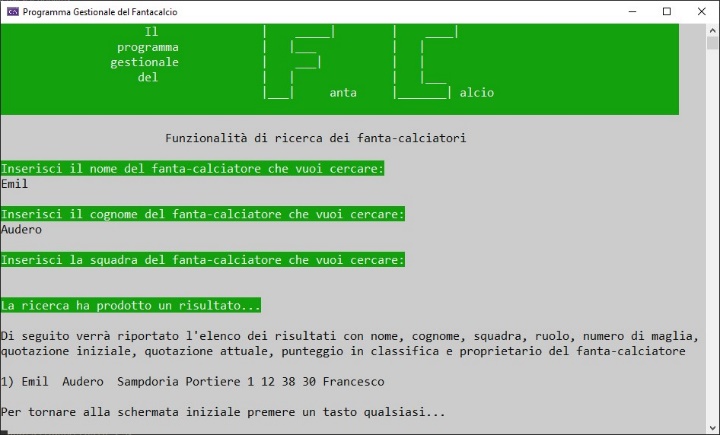
In questo caso l’utente, una volta inserito il numero che corrisponde alla funzione in elenco, non ha bisogno di premere **invio** perché il programma effettua la conferma automaticamente.

**Schermata di inserimento delle rose dei giocatori**

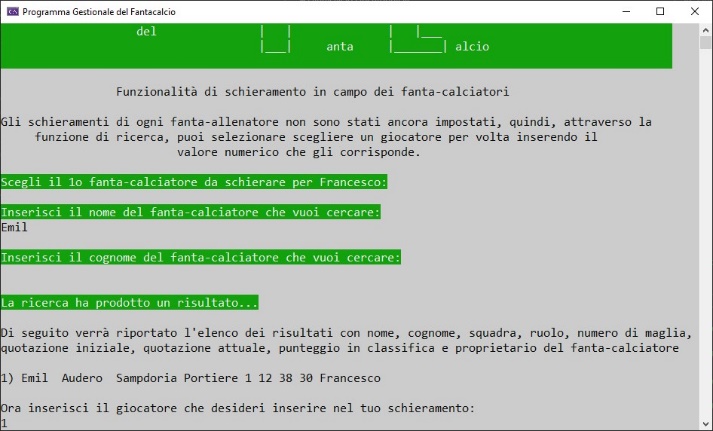
La schermata di inserimento delle rose dei giocatori (immagine a fianco) non è attivabile direttamente dalla schermata iniziale con un comando in caso di avvio comune, ma solamente nel caso di un primo avvio.   
Supponendo che i fanta-allenatori si siano procurati manualmente le liste dei fantagiocatori e che abbiano già provveduto a fare l'asta, il programma chiederà l’inserimento delle rose dei giocatori: da parte del software verrà effettuata una suddivisione automatica in base ai ruoli (portiere, difensore, centrocampista e attaccante), quindi l'utente dovrà inserire manualmente il nome, il cognome, la squadra di appartenenza, il numero di maglia e la quotazione iniziale del fanta-calciatore, che devono essere obbligatoriamente inseriti separati dal carattere virgola.  
Poi, quando ogni fanta-allenatore avrà inserito tutti i suoi fantagiocatori, le rose verrano riepilogate. Alla fine verrà chiesto all’utente se vuole effettuare il salvataggio o meno: nel primo caso inserisce “S”, viene effettuato il salvataggio e il programma porta l’utente alla schermata iniziale, altrimenti inserisce “N” e il programma si chiude.

**Schermata di visualizzazione dei dati dei fanta-allenatori**

Attraverso la schermata iniziale è possibile accedere alla schermata di visualizzazione dei dati dei fanta-allenatori (quella dell’immagine a fianco): qui vengono mostrati nome e budget disponibile di ogni fanta-allenatore inserito.   
Una volta finita la consultazione si può tornare alla schermata iniziale premendo un tasto qualsiasi.

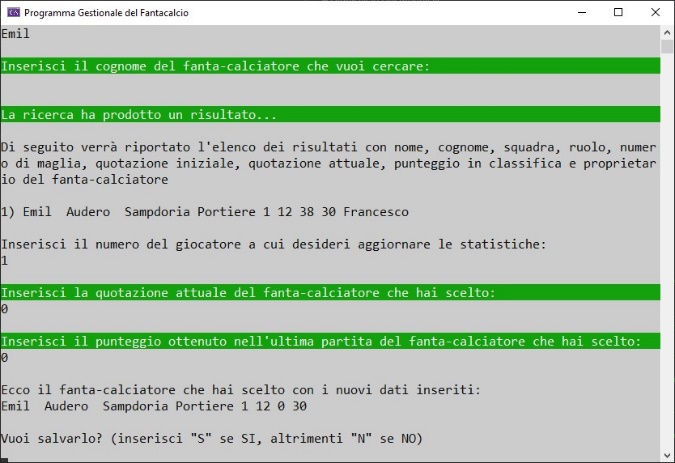
**Schermata di ricerca/visualizzazione dei   
fanta-calciatori**

Una volta che tutti i passaggi iniziali sono stati completati, tramite la schermata iniziale è possibile accedere alla schermata dei fanta-calciatori. Qui è possibile eseguire una ricerca dei fanta-calciatori inserendo, come filtri, le caratteristiche dei giocatori: nome, cognome, squadra, numero di maglia, ruolo, quotazione iniziale, quotazione attuale e punteggio in classifica.  
Tutti i filtri, tranne il nome, si possono saltare non inserendo nulla e premendo direttamente **invio**: in questo caso viene subito mostrato il numero di risultati della ricerca. Successivamente, se la ricerca ha ottenuto uno o più risultati, viene visualizzata la lista con tutti i fantagiocatori che corrispondo ai filtri, insieme alle loro caratteristiche e il nome del fanta-allenatore che li possiede (come nell’immagine a fianco). I risultati di ricerca, poi, vengono visualizzati a gruppi di diciassette, in modo che la lettura risulti più agile per l’utente: per passare a un gruppo successivo basta premere un tasto qualsiasi.  
Una volta finita la consultazione si può tornare alla schermata iniziale premendo un tasto qualsiasi.

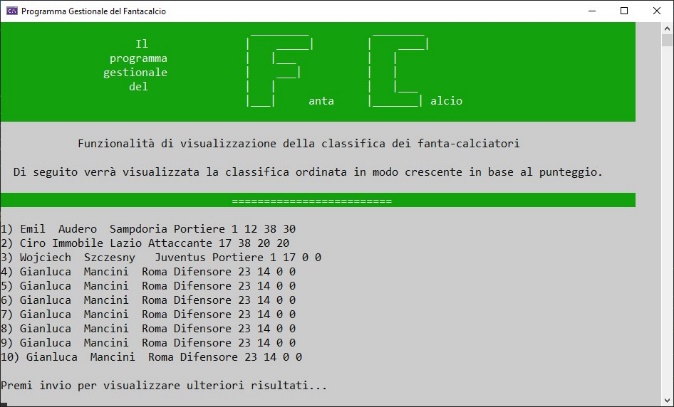
**Schermata di schieramento in campo**

Sempre dopo aver completato l’inserimento dei dati iniziali, è possibile accedere alla schermata di schieramento in campo. Come prevede il regolamento del fantacalcio, è necessario che ogni fanta-allenatore schieri in campo undici giocatori ogni qualvolta che avvenga una partita reale, più eventuali riserve, che vengono scelti dalle proprie rose. Anche qui le situazioni che si possono presentare sono due.

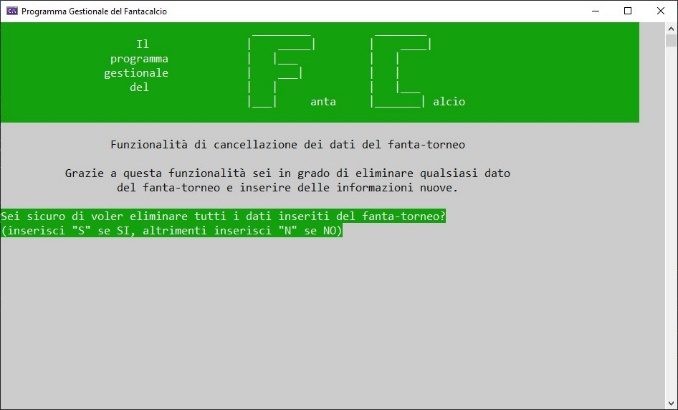
* Se gli schieramenti non sono ancora stati impostati, allora è necessario effettuare una ricerca per ognuno dei fanta-calciatori che si vuole schierare e sceglierne uno che si trova nell’elenco dei risultati inserendo il numero che gli corrisponde (come nell’immagine a fianco); se, invece, la ricerca non produce alcun risultato, allora viene chiesto all’utente se desidera riprovare a cercare un fanta-calciatore oppure vuole tornare alla schermata iniziale: nel primo caso inserisce “**S**”, mentre nel secondo inserisce “**N**”.  
  Una volta che un fanta-giocatore è stato scelto per lo schieramento, le stesse operazioni precedentemente spiegate vengono ripetute fino al completamento degli schieramenti di tutti i fanta-allenatori. Alla fine di tutto viene chiesto all’utente se vuole salvare gli schieramenti: può rispondere con “**S**” per confermare, oppure “**N**” per tornare alla schermata iniziale.
* Se lo schieramento è stato già impostato, invece, il programma lo visualizza e chiede all’utente se vuole cambiarlo oppure se vuole tornare alla schermata iniziale: nel primo caso l’utente inserisce “**N**” e viene effettuata l’eliminazione degli schieramenti, altrimenti se inserisce “**S**” il programma ritorna alla schermata iniziale.

**Schermata di aggiornamento delle statistiche**

La schermata di aggiornamento delle statistiche (immagine a fianco) permette di aggiornare i punteggi di ogni fanta-giocatore al termine di ogni partita reale del torneo scelto. Attraverso la funzione di ricerca già spiegata prima, si cerca un fantagiocatore, si inserisce il numero a cui corrisponde nell’elenco dei risultati, si inserisce la quotazione attuale e, in base ai bonus/malus scelti, si aumenta (inserendo +*punteggio*) o se ne diminuisce il punteggio (inserendo -*punteggio*). Una volta inserito il nuovo punteggio, il programma chiede se si vuole salvare la nuove statistiche. L’utente può rispondere positivamente con “**S**”, oppure negativamente con “**N**”.   
In qualunque caso, comunque, il software chiede all’utente se desidera aggiornare le statistiche di un altro fanta-giocatore, oppure tornare alla schermata iniziale: anche qui, nel primo caso inserisce “**S**” mentre nel secondo inserisce “**N**”.

**Schermata di visualizzazione delle classifiche parziali**

Questa schermata (immagine a fianco) permette di visualizzare la classifica parziale ordinata in modo decrescente in base al punteggio ottenuto da ogni fanta-calciatore nel torneo di Fantacalcio.  
Così come nella ricerca, la classifica non viene visualizzata immediatamente in tutti i suoi elementi, ma a gruppi di dieci: prima di passare ad un altro gruppo è necessario che l’utente prema un tasto qualsiasi. Dopo aver completato la lettura, si può tornare alla schermata iniziale premendo un tasto qualsiasi.

**Schermata di cancellazione dei dati**

La schermata di cancellazione dei dati (immagine a fianco) permette di far ritornare il programma allo stato iniziale, cancellando ogni dato inserito riguardo al torneo di Fantacalcio: questo tipo di operazione può rendersi necessario nel caso in cui il torneo sia terminato naturalmente oppure per decisione dei fanta-allenatori.   
Il programma chiederà la conferma all’utente se vuole proseguire l’operazione di eliminazione: in caso affermativo inserisce “**S**”, cancella i dati e riavvia l’applicazione, che visualizzerà una schermata iniziale da primo avvio, altrimenti inserisce “**N**” e può tornare alla schermata iniziale.

**Uscire dal programma**

Il software può essere chiuso dalla schermata iniziale in caso di avvio comune con il comando numerico “**7**”, oppure con l’icona presenta nella finestra: se si esce dal programma in quest’ultimo modo mentre sono in esecuzione operazioni di salvataggio, oppure nel caso in cui l’utente non abbia terminato un inserimento, allora si rischia di perdere i dati, visto che il salvataggio non viene effettuato per ogni inserimento.

**File e gestione degli errori**

Precedentemente quando si faceva riferimento al salvataggio dei dati, o qualunque altro tipo di modifica che veniva effettuata a essi, si intendeva dire, dal punto di vista pratico, che viene effettuata una serie di operazioni sui file di salvataggio: il loro funzionamento sarà approfondito nell’analisi tecnica.   
Ad ogni modo, ogni volta che viene effettuata questa serie di procedure sui file possono verificarsi degli errori: questi vengono gestiti e la descrizione dell’errore che si verifica viene sempre visualizzato sulla console, quindi l’utente ne è sempre consapevole.

**Controlli di inserimento corretto delle informazioni**

In diverse funzioni del programma vengono eseguiti diversi controlli informatici sul tipo di dato che viene inserito e con che valore, pertanto, non si dovrebbero verificare errori di funzionamento del programma.   
Per quanto riguarda l’attinenza delle informazioni inserite rispetto al fanta-torneo e al regolamento del Fantacalcio, invece, in molti casi si lascia la verifica all’utente: per ragioni di tempo non è stato possibile includere questo genere di test.

**Analisi tecnica**

**Premessa**  
Il programma ha un’interfaccia grafica di tipo CLI (Command Line Interface), realizzata con .NET Core 3.1, attraverso l’uso del linguaggio di programmazione C#. Per realizzare il programma si è proceduto a usare l’ambiente di sviluppo Microsoft Visual Studio 2019. Di seguito verrà spiegata buona parte del funzionamento tecnico del programma; per analizzare ulteriormente la parte tecnica consultare la documentazione interna (commenti) all’interno del codice del programma.

**Direttive *using***

Le direttive *using*, o spazio dei nomi, che sono stati utilizzati sono i seguenti:

* *using* **System**, direttiva che permette l’uso dei metodi più comuni;
* *using* **System.IO**, fondamentale per le operazioni sui file;
* *using* **System.Collections.Generic**, necessaria per la creazione gestione di liste;
* *using* **Newtonsoft.JSON**, che permette l’uso di metodi specifici per l’utilizzo di file JSON.

**Le liste**

Per contenere le informazioni sul fanta-torneo durante l’esecuzione del programma, si usano le liste (List): List è una classe che rappresenta un elenco di oggetti fortemente tipizzato accessibile per indice; fornisce metodi per la ricerca, l'ordinamento e la modifica degli elenchi.   
All’interno del software si fa uso di liste che hanno un tipo di oggetti “standard”, come integer, mentre altre considerano come oggetti gli attributi di una classe: per esempio la lista FantaAllenatori ha come elementi di una “cella” nome, codice rosa e budget disponibile di un fanta-allenatore.

**Gestione dei file JSON**

Una particolarità del software consiste nell’utilizzo di file di salvataggio di tipo **JSON**: questo tipo di file consente di memorizzare in modo più ordinato e coerente le informazioni rispetto al semplice file di testo (con estensione .**txt**). Un file JSON si presenta con un formato del tipo:

{

"***nome attributo***": "***attributo***",

… (*altri attributi*)

}, … (*altro insieme di attributi*)

La operazioni su un file JSON vengono effettuate sempre con i metodi di ***using* System.IO** (come **File.ReadAllText()**).   
All’interno del *Programma Gestionale del Fantacalcio* sono stati poi utilizzati fondamentalmente due tipi di metodi ad hoc per i file JSON, resi disponibili dallo spazio dei nomi ***using* Newtonsoft.JSON**:

* si è effettuata la **serializzazione** dei dati, cioè la trasformazione degli elementi presenti all’interno delle liste di fanta-allenatori, fanta-calciatori e dei codici di schieramento in un formato ben preciso che può essere inserito all’interno di un’unica stringa (**JsonConvert.SerializeObject()**);
* si è effettuata la **deserializzazione** dei dati, un processo inverso a quello precedente, dove da una stringa in ingresso in formato JSON vengono estratte le informazioni che possono essere inserite all’interno delle liste dei fanta-allenatori, fanta-calciatori e dei codici di schieramento (**JsonConvert.DeserializeObject()**).

**Paradigma di programmazione**

Il codice verrà redatto seguendo la **OOP**, ovvero la programmazione orientata agli oggetti, che prevede la presenza di:

* ***classi***, che hanno il compito di definire le caratteristiche di un insieme di oggetti che hanno proprietà e compiono azioni uguali;
* ***attributi***, cioè delle variabili e/o costanti che definiscono le caratteristiche o proprietà degli oggetti instanziabili invocando la classe; i valori inizializzati degli attributi sono ottenuti attraverso il cosiddetto costruttore;
* ***metodi***, ossia procedure che operano sugli attributi.

Gli oggetti, quindi, sono istanze delle classi.

**Classi**

Le classi che compongono il codice del programma, insieme ai loro metodi e attributi, sono le seguenti.

* Classe **Programma**: questa classe viene eseguita automaticamente per prima all’avvio del programma.   
  Questa classe non ha attributi. Le variabili principali della classe sono:
  + **percorsiFile** (vettore unidimensionale di string), che contiene i percorsi dei file di salvataggio;
  + **percorsoCartella** (string), che contiene il percorso della cartella di salvataggio;
  + **FantaAllenatori** (lista che ha come elementi gli attributi della classe FantaAllenatore), che contiene i dati dei fanta-allenatori;
  + **FantaCalciatori** (lista che ha come elementi gli attributi della classe FantaCalciatore), che contiene i dati dei fanta-calciatori;
  + **CodiciSchieramenti** (lista che ha due elementi di tipo integer per cella), gli indici nella lista FantaCalciatori dei fanta-calciatori schierati insieme al codice della rosa di cui fanno parte;
  + **Indici** (lista che ha un elemento di tipo integer per cella), che contiene gli indici nella lista FantaCalciatori dei fanta-calciatori che sono stati trovati da una ricerca.

I metodi della classe sono:

* **Main**, il metodo a cui il programma fa per primo riferimento in seguito al suo avvio, che contiene fondamentalmente le operazioni di preparazione dei dati per il loro utilizzo e il riferimento ai metodi che permettono la visualizzazione della schermata iniziale (**PrimoAvvio**, **AvvioComune**);
* **VisualizzaIntestazione**, che permette di visualizzare l’intestazione di ogni schermata, ovvero la scritta “Programma gestionale del Fantacalcio”;
* **PrimoAvvio**, richiamato dal Main solamente quando il programma viene avviato per la prima volta; contiene le visualizzazioni video e le prime richieste di inserimento per l'utente;
* **SchermataInserimentoRose**, che viene richiamato solo da PrimoAvvio e che contiene la visualizzazione video delle richieste che permettono all'utente l'inserimento delle rose di fanta-calciatori;
* **AvvioComune**, metodo richiamato dal Main quando i dati iniziali sono stati configurati; permette la visualizzazione del menu di scelta tra le diverse funzionalità del programma;
* **SchermataRicercaFantaCalciatori**, che contiene la parte di visualizzazione a schermo della schermata di ricerca dei fanta-giocatori;
* **SchermataVisualizzazioneFantaAllenatori**, che permette di visualizzare i dati dei fanta-allenatori;
* **SchermataSchieramentoCampoFantaCalciatori**, che permette di creare o visualizzare gli schieramenti in campo dei fanta-calciatori;
* **RispostaSiNo**, che permette di effettuare l'inserimento e verificare la risposta di un utente nel caso in cui si richieda che essa sia "S" o "N";
* **SchermataAggiornamentoStatisticheFantaCalciatori**, che permette di visualizzare le richieste all'utente che porteranno all'aggiornamento delle statistiche dei fanta-calciatori;
* **Ricerca**, che permette di visualizzare le richieste dei filtri e i risultati della ricerca;
* **SchermataVisualizzazioneClassifiche**, che contiene la parte di visualizzazione a schermo della schermata delle classifiche parziali dei fanta-giocatori;
* **SchermataCancellazioneDati**, che contiene la parte di visualizzazione a schermo della schermata di cancellazione dei dati;
* **SetResetColori**, che permette di effettuare, in base al valore booleano che gli viene fornito come argomento, il cambio dei colori dello sfondo e dei caratteri visualizzati, oppure riportare i colori alle impostazioni standard predefinite.
* Classe **OperazioniFile**: questa classe contiene tutti i metodi necessari per eseguire la lettura e scrittura dei files. Questa classe non ha attributi e le variabili necessarie al funzionamento dei metodi vengono passate ad ognuno di essi nella loro invocazione.

I metodi della classe sono:

* **VerificaEsistenzaFile**, metodo che permette di verificare se esiste o meno il percorso del file che gli è stato passato attraverso **File.Exists()**;
* **CreaCartella**, metodo che viene richiamato quando la cartella di salvataggio non è presente; contiene il metodo **Directory.CreateDirectory()**, le cui eccezioni verranno gestite attraverso un **blocco try-catch**;
* **LeggiFile**, metodo che permette, attraverso **File.ReadAllLines()**, la lettura di un file di salvataggio;
* **ScriviFile**, metodo che permette di scrivere in un file di testo attraverso **File.WriteAllLines()**, le cui eccezioni verranno gestite attraverso un **blocco try-catch**;
* **EliminaFile,** metodo che permette di eliminare un file attraverso **File.WriteAllText()**, le cui eccezioni verranno gestite attraverso un **blocco try-catch**;
* **GestisciErrori**, metodo richiamato nel caso di eccezioni verificatasi nei metodi precedentemente spiegati, che permette di restituire messaggi di errore da visualizzare nelle varie situazioni che si potrebbero verificare.
* Classe **FantaAllenatore**: questa classe permette di accedere ai metodi che riguardano i fanta-allenatori.   
  Gli attributi della classe sono:  
  + **nome** (string), ovvero il nome del fanta-allenatore;
  + **codiceRosa** (string), che indica il numero di rosa che appartiene al fanta-allenatore;
  + **budgetDisponibile** (integer), che indica il numero di Fantamilioni che sono rimasti al fanta-allenatore.

I metodi della classe, invece, sono:

* Metodo **costruttore**, che permette l’effettivo passaggio dei valori forniti come argomento nell’istanziazione della classe agli attributi della classe corrente;
* **AggiungiFantaAllenatore**, che permette di aggiungere un fanta-allenatore nella lista FantaAllenatori;
* **OttieniNome**, che permette di accedere al nome di un fanta-allenatore.
* Classe **FantaCalciatore**: questa classe permette di accedere ai metodi che riguardano i fanta-calciatori. Gli attributi della classe sono:
  + **nome** (string), ovvero il nome del fanta-calciatore;
  + **cognome** (string), ovvero il cognome del fanta-calciatore;
  + **squadra** (string), ovvero la squadra di appartenenza del fanta-calciatore;
  + **ruolo** (string) , ovvero il ruolo fanta-calciatore all’interno della squadra;
  + **numeroMaglia** (integer), ovvero il numero di maglia del fanta-calciatore;
  + **quotazioneIniziale** (integer), ovvero la quotazione iniziale del fanta-calciatore;
  + **quotazioneAttuale** (integer), ovvero la quotazione attuale del fanta-calciatore;
  + **punteggioClassifica** (integer), ovvero il punteggio del fanta-calciatore nella classifica parziale;
  + **codiceRosa** (integer), ovvero il codice della rosa a cui un fanta-calciatore appartiene.

I metodi della classe, invece, sono:

* Metodo **costruttore**, che permette l’effettivo passaggio dei valori forniti come argomento nell’istanziazione della classe agli attributi della classe corrente;
* **AggiungiFantaCalciatore**, metodo che permette di aggiungere un giocatore alle liste di fanta-calciatori;
* **RicercaFantaCalciatore**, che permette di ricercare uno o più fanta-giocatori in base ai filtri inseriti dall’utente grazie a List<T>.FindAll();
* **DefinisciPredicato**, richiamato dal metodo precedente che definisce il predicato (Predicate <T>) che deve essere usato dal metodo List<T>.FindAll() per effettuare la comparazione tra i diversi elementi di una lista;
* **OrdinaFantaCalciatori**, che ordina in modo decrescente i giocatori in base al loro punteggio, utilizzando il metodo List<T>.Sort();
* **ComparaPunteggi**, che definisce il tipo di metodo Comparison <T> che deve essere usato dal metodo List<T>.Sort() per l’ordinamento della lista ;
* **OttieniCodiceRosa**, che permette di accedere al codice della rosa in cui si trova il fanta-calciatore;
* **AggiornaQuotazioneAttuale**, che permette di accedere alla quotazione attuale di un fanta-calciatore e modificarla;
* **AggiornaPunteggio**, che permette di accedere alla punteggio in classifica di un fanta-calciatore e aggiornarlo, effettuando la somma del punteggio attuale e quello della partita che si è appena svolta.

Per concludere, in generale:

* ogni classe, tranne Programma, contiene il metodo di tipo override (un metodo cioè che ne sovrascrive un altro già esistente) **ToString**, e che ritorna il valore di una stringa costruita da più stringhe, pronta per essere visualizzata a schermo.
* ogni metodo, presenta delle variabili interne necessarie ai fini del funzionamento del programma, per esempio indici o contatori, che non possono essere completamente elencati all’interno di questa analisi.

**Conclusioni**

Nel complesso il progetto si è svolto correttamente: la consegna è stata posticipata di circa una settimana rispetto alla tempistica inizialmente stabilita, ma la maggior parte delle caratteristiche preventivate nelle analisi iniziali sono state sviluppate interamente. È necessario aggiungere, però, che se il progetto dovesse essere continuato si potrebbero fare diversi miglioramenti innanzitutto all’usabilità del programma: si potrebbe implementare maggiori controlli sui dati inseriti dall’utente in modo da far rispettare pienamente il regolamento del fantacalcio, ma anche fare in modo che le richieste, in particolare quelle di inserimento di dati, possano essere realizzate più velocemente, andando maggiormente incontro alle necessità dell’utente, che desidera gestire i dati in modo efficiente senza perdere troppo tempo. Infine, potrebbe essere d’aiuto per chi usa il programma l’aggiunta di ulteriori funzionalità di gestione del fanta-torneo: per esempio, potrebbe essere considerato il fatto che un giocatore può essere trasferito da una squadra all’altra, quindi si renderebbe necessaria una gestione più complessa sia delle rose che dei crediti di cui dispongono i fanta-allenatori.

**Fonti della relazione**

* [https://www.focusjunior.it/news/news-e-curiosita/Fantacalcio-cose-e-come-si-gioca/](https://www.focusjunior.it/news/news-e-curiosita/fantacalcio-cose-e-come-si-gioca/)
* [https://www.Fantacalcio.it/consigli-Fantacalcio/14\_10\_2020/consigli-Fantacalcio-come-dividere-il-budget-all-asta-391828](https://www.fantacalcio.it/consigli-fantacalcio/14_10_2020/consigli-fantacalcio-come-dividere-il-budget-all-asta-391828)
* <https://it.wikipedia.org/wiki/Fantacalcio>
* <https://dotnet.microsoft.com/download/dotnet/3.1>
* <https://docs.microsoft.com/it-it/dotnet/csharp/>
* <https://it.wikipedia.org/wiki/Programmazione_orientata_agli_oggetti>
* <https://csharp.net-tutorials.com/it/389/collections/lists/>