

# PROGETTO IUM-TWEB

*~ Football Whispers ~*

*<https://github.com/ZakAbourida/IUM-TWEB-UniTo.git>*



**Zakaria Abourida - 950120**

**Daniel Rabottini - 933913**

**Tommaso Remondino - 947410**

2023/2024

# INTRODUZIONE

In risposta ai requisiti delineati nella traccia del progetto, abbiamo sviluppato diverse pagine HTML, ognuna focalizzata su una specifica sezione delle informazioni estratte dalle collezioni di dati:

- **Homepage:** Offre informazioni generali, quali notizie, prossime partite e risultati delle ultime partite disputate.
- **Canale:** Presenta un elenco di canali dedicati, fornendo la possibilità agli utenti di discutere su vari argomenti.
- **Lista Campionati:** Visualizza l'elenco dei campionati disponibili.
- **Lista Squadre:** Mostra tutte le squadre presenti nel campionato selezionato, insieme alle informazioni di base sul campionato e ai principali giocatori.
- **Squadra:** Una pagina dedicata alla squadra selezionata, che fornisce informazioni basilari, una lista dei giocatori associati alla squadra, lo storico delle partite e una sezione dedicata alle statistiche della stagione.
- **Lista Giocatori:** Offre una pagina per la ricerca "avanzata" dei giocatori, con filtri basati su campionato, nazione, anno, e altri parametri.
- **Giocatore:** Una pagina specifica per ogni giocatore selezionato, comprendente statistiche, informazioni di base e altri dettagli rilevanti.

Per l'architettura web, abbiamo creato un server principale con Express che gestisce le chiamate e comunica con due server dedicati per l'accesso ai database: uno per MongoDB (sempre Express), uno per PostGres (Java Spring Boot) e uno Python Flask (per la generazione dei grafici). Questo approccio permette una gestione efficiente delle richieste per ciascun database, assicurando un'architettura solida e organizzata.

## DOCUMENTAZIONE TASK

### TASK 1: Realizzazione delle pagine HTML (con Javascript e CSS)

- **Soluzione:** Il design e la struttura delle pagine sono stati influenzati dai risultati ottenuti nel progetto di IUM. Abbiamo optato per la palette di colore viola/giallo al fine di differenziarci, considerando che la maggior parte dei siti di calcio utilizza una palette comune di grigio, bianco, nero e verde. Riguardo alla struttura, ci siamo ispirati ai principi di accessibilità e usabilità appresi nel corso di IUM, cercando di garantire un'esperienza utente ottimale.
- **Problemi:** La complessità di questa sezione è stata in gran parte autoimposta; la scelta di alcuni elementi, come i menù a tendina e le tabelle scorrevoli, ha presentato sfide nella loro implementazione completa e funzionale.
- **Requisiti:** Riteniamo di aver soddisfatto tutti i requisiti, poiché sebbene non vi fossero condizioni specifiche, abbiamo goduto di una certa flessibilità nell'approccio alla loro

implementazione.

- **Limitazioni**: Non abbiamo incontrato limitazioni significative, poiché eravamo abbastanza liberi di sperimentare e seguire la nostra intuizione, a condizione che il funzionamento complessivo fosse efficiente. La nostra soluzione è altamente estendibile, la sua scalabilità dipende unicamente dalla quantità e dalla tipologia dei dati nelle collezioni e dalla loro modalità di visualizzazione. Inoltre sono stati lasciati apposta delle possibili “features” per una possibile implementazione futura.

## **TASK 2: Implementazione server e middleware**

- **Soluzione**: La nostra soluzione si compone di un server principale implementato tramite Express, un altro server Express dedicato al database MongoDB, un server Java Spring Boot per il database PostGres ed infine un server Python Flask. Il server principale agisce come intermediario, ricevendo le chiamate Axios dal client e successivamente reindirizzandole al server del database appropriato. Quest'ultimo esegue le query necessarie nel database e restituisce i dati richiesti. I server connessi ai database gestiscono efficacemente queste comunicazioni con il server principale e gestiscono l'accesso ai rispettivi database. Il server Python Flask permetta la visualizzazione di grafici dedicati nella pagina Giocatore.
- **Problemi**: Questa sezione si occupa della gestione e dell'applicazione delle metodologie di comunicazione tra i vari server, nonché della gestione dei dati. La sfida principale è emersa durante la comprensione dettagliata del funzionamento dell'intera struttura in base alle nostre esigenze e nella selezione di strategie per il trasferimento efficace dei dati. Le difficoltà sono state evidenti soprattutto nelle prime fase di creazione delle prime funzioni e query, richiedendo una comprensione approfondita e la risoluzione di eventuali errori durante lo sviluppo.
- **Requisiti**: Crediamo di aver adempiuto a tutti i requisiti poiché il server principale, sviluppato con Express, gestisce le comunicazioni in modo rapido ed efficiente, mentre i server dei database (Express e Java Spring Boot) eseguono correttamente le funzioni di query e rispondono al server principale, rispettando così le specifiche richieste.
- **Limitazioni**: Crediamo che la nostra soluzione sia facilmente estendibile a seconda delle informazioni e dal contesto specifico. Possiamo aggiungere numerosi nuovi tipi di query e funzioni per migliorare la visualizzazione dei dati.

## **TASK 3: Suddivisione collezioni di dati e gestione dei database**

- **Soluzione**: La direttiva richiede di separare le collezioni di dati in due database, distinti in dati statici e dinamici. Abbiamo scelto di inserire nel database MongoDB i dati dinamici: presenze, partite del club, eventi di gioco, formazioni di gioco e partite.

Nel database PostGres, invece, abbiamo inserito i dati statici, ovvero: giocatori, club, competizioni e valutazioni dei giocatori.

- **Problemi**: L'unico ostacolo che abbiamo incontrato è stato capire come importare le collezioni nei rispettivi database. Tuttavia, con questa suddivisione, non si sono verificati problemi nella visualizzazione dei dati. Le query, principalmente di natura semplice o comunque le più complesse che richiedevano maggiori informazioni, trovavano tutte le risorse necessarie nei database separati.
- **Requisiti**: Riteniamo che la soluzione soddisfi i requisiti richiesti dalla direttiva.
- **Limitazioni**: Riteniamo che all'interno delle collezioni di dati, soprattutto per quelle dinamiche, come le informazioni sulle partite e gli eventi, siano presenti poche informazioni dettagliate. Nello specifico, riguardo alle partite, sono rappresentati solo i gol, gli assist, i cartellini rossi e gialli, e le sostituzioni. Questa limitatezza ci ha reso difficoltosa la visualizzazione "completa" di alcune statistiche. L'aggiunta di dati come i tiri in porta e i passaggi avrebbe consentito di mostrare un set più ampio di statistiche e derivare ulteriori informazioni, come il possesso palla, la percentuale di tiri e la dominanza in partita.

#### **TASK 4: Sistema delle chat (pagina Canale)**

- **Soluzione**: Dopo aver inserito l'utente e la password, le credenziali vengono salvate per consentirne l'utilizzo nella pagina Canale. Qui abbiamo creato quattro canali, ognuno dedicato a un argomento specifico. Una volta selezionato un canale, si accede alla stanza dedicata, consentendo agli utenti di chattare con altri partecipanti connessi alla stessa stanza. Tutto ciò è reso possibile attraverso l'utilizzo di Socket.io.
- **Problemi**: La direttiva chiedeva di implementare un sistema di chat per permettere agli utenti di discutere su vari argomenti... Le principali sfide sono state la gestione dell'elenco degli utenti collegati alla stessa stanza e il controllo sull'username degli utenti.
- **Requisiti**: Riteniamo di aver adempiuto ai requisiti richiesti, poiché non sono state specificate richieste particolari.
- **Limitazioni**: Naturalmente, la nostra soluzione è altamente espandibile; ad esempio, potrebbe includere una gestione più avanzata degli username, l'archiviazione completa delle conversazioni e la possibilità di inserire immagini, GIF e altri elementi extra-testuali.

#### **TASK 5: Analisi dei dati su larga scala con Jupyter Notebook**

- **Soluzione**: Abbiamo adottato un metodo di analisi top-down, esplorando sistematicamente i dataset per garantire chiarezza e accuratezza nell'approccio. Ogni dataset è stato dettagliatamente esaminato in sezioni separate, con sottosezioni che

consentono un'analisi approfondita di specifici aspetti. L'obiettivo principale è stato identificare tendenze generali nel mondo del calcio, pur affrontando casi particolari legati a singoli giocatori o squadre dove rilevante. Questo approccio ci ha permesso di scoprire modelli e insight che altrimenti sarebbero rimasti nascosti.

- **Problemi**: Il principale ostacolo è stato l'interpretazione dei dati nei dataframe e la formulazione di strategie di analisi che fossero non solo significative per noi come autori, ma anche facilmente comprensibili per altri lettori. Questo ha richiesto un attento equilibrio tra tecnicismo e accessibilità nell'esposizione dei risultati.
- **Requisiti**: La nostra analisi rispetta ampiamente i criteri richiesti per l'assignment. Sebbene ci siano margini di miglioramento, siamo convinti che il lavoro svolto fornisca una solida base di partenza, mostrando un'attenta considerazione dei requisiti e degli obiettivi del progetto.
- **Limitazioni**: Una limitazione rilevante nel nostro approccio analitico con Python è stata l'incapacità iniziale di generare grafici interattivi basati su parametri dinamici. Questo limite, dovuto principalmente alla nostra esperienza iniziale limitata, è stato ora riconosciuto e stiamo lavorando per implementare soluzioni che permettano maggiore flessibilità e interattività nelle visualizzazioni dei dati, migliorando così significativamente la qualità dell'analisi e della presentazione dei risultati.

## CONCLUSIONI

Il percorso di progettazione della soluzione si è rivelato un'esperienza formativa di grande rilievo, consentendoci di mettere in pratica i concetti e le metodologie acquisite durante il corso. Questi ci hanno fornito una comprensione più approfondita delle architetture web e delle dinamiche sottostanti a una pagina web. Inoltre, abbiamo apprezzato l'importanza e la complessità nell'organizzazione di un progetto di tale portata. Le sfide principali sono state affrontate nell'organizzazione complessiva del progetto e nella sincronizzazione delle attività di sviluppo tra di noi. Questa esperienza ci ha fornito preziose lezioni sulla collaborazione e sull'organizzazione necessarie per portare a termine progetti estesi e complessi.

## DIVISIONE DEI LAVORI

Abourida Zakaria: pagine HTML(con javascript e css), backend MongoDB, documentazioni Swagger

Daniel Rabottini: Jupyter Notebook, backend PostGres, Python Flask

Tommaso Remondino: pagine HTML(con javascript e css), Sistema chat

Nonostante la ripartizione dei compiti, abbiamo mantenuto una collaborazione costante tra di

noi per ciascuna sezione, contribuendo ciascuno in modo significativo in tutte le sezioni del progetto.

## INFORMAZIONI EXTRA

Abbiamo sfruttato l'assistenza di ChatGpt principalmente per il debugging e la risoluzione di eventuali problemi sorti durante lo sviluppo. Inoltre, ci è stato di aiuto per migliorare e ottimizzare alcune funzioni, contribuendo in modo significativo al perfezionamento complessivo della nostra soluzione, con un focus particolare sulla parte relativa al CSS.

Linee guida per far partire la simulazione:

1. Creare un database 'DynamicDatabase' con MongoCompass e creare le collezioni (appearances, game\_events, game\_lineups, games, club\_games). Successivamente importare i file CSV per popolare il database.

2. Far partire i server:

1. Frontend/bin/www (porta: 3000)

2. Backend/serverMongo/bin/www (porta: 3001)

3. Backend/servePostgres/src/main/java/ium/tweb/serverpostgres/ServerPostgresApplication (porta: 8081)

4. Backend/flaskServer/app.py (porta: 127.0.0.1:5000)

3. Iniziare la simulazione da localhost:3000

## BIBLIOGRAFIA

Abbiamo utilizzato principalmente le slides e porzioni di codice provenienti dalle lezioni e laboratori. Inoltre abbiamo fatto uso di siti web d'informazione come Stack Overflow, w3schools e Bootstrap.

Abbiamo utilizzato un API per ottenere un font specifico che avevamo utilizzato nella progettazione del sito nella parte di IUM.

