

UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI ECONOMIA

Corso di Laurea in DATA SCIENCE PER L'ECONOMIA E LE IMPRESE

L'impatto degli strumenti digitali e dei data analytics nelle prestazioni sportive: Il caso A.C. Perugia Calcio

The impact of digital tools and data analytics on sports performance: The A.C. Perugia Calcio case

Relatore

Prof. Emanuele Storti

Laureando

Giacomo Famiani

A.A. 2025/2026

"A chi c'era prima dei libri e degli esami:
Al mio primo amore,
a quel pallone fatto di esagoni
che mi ha dato la forza di arrivare fin qui."

Indice

Introduzione 8

CAPITOLO 1 12

1.1 OBIETTIVO GENERALE DELLA TESI 12

1.2 PROBLEMATICHE APERTE NEL DOMINIO DI RIFERIMENTO 14

1.3 ANALISI E DEFINIZIONE DEI REQUISITI 15

1.3.1 REQUISITI PER LA PIPELINE DI ANALISI TATTICA E REPORTISTICA (Il Caso A.C. Perugia Calcio) 15

1.3.2 REQUISITI PER IL SISTEMA GESTIONALE FULL-STACK (gestionale_asd_calcio) 16

CAPITOLO 2 19

2.1 LINGUAGGI DI PROGRAMMAZIONE & CORE TECHNOLOGIES 19

2.1.1 PYTHON (V3.X) 19

2.1.2 R STUDIO 20

2.1.3 HTML, CSS e JAVASCRIPT 21

2.2 FRAMEWORK DI SVILUPPO WEB E INTERFACCE API 23

2.2.1 DJANGO 23

2.2.2 INTERFACCE API (Application Programming Interfaces) 24

2.3 INTEGRAZIONE DI MODELLI DI LINGUAGGIO (LLM) 25

2.3.1 GEMINI PRO (GOOGLE GenAI) 25

2.4 BUSINESS INTELLIGENCE, VISUALIZZAZIONE E FOGLI DI CALCOLO 26

2.4.1 POWER BI 26

2.4.2 MICROSOFT EXCEL 27

2.5.1 MYSQL e MYSQLite 28

2.5.2 PANDAS e NUMPY 29

2.6 AMBIENTI DI SVILUPPO, CONTROLLO DI VERSIONE e TRACCIABILITÀ 31

2.6.1 VISUAL STUDIO CODE (VS CODE) 31

2.6.2 GIT e GITHUB 32

CAPITOLO 3 33

3.1 INTRODUZIONE E ARCHITETTURA DELLA PIPELINE DEI DATI 33

3.2 INGEGNERIA DEL DATO: FONTI ETEROGENEE, STRUTTURA E VALIDAZIONE DEL DATASET. 36
3.2.1. L'ECOSISTEMA DELLE FONTI E I CANALI DI INGESTIONE 36
3.2.2 NORMALIZZAZIONE, SINCRONIZZAZIONE E DATA WRANGLING TRAMITE SCRIPT PYTHON 38
3.2.3 LA VALIDAZIONE DI DOMINIO CON LO STAFF TECNICO 39
3.2.4 MODELLAZIONE RELAZIONALE ED INTEGRAZIONE IN POWER BI 40
3.3 MODELLAZIONE STATISTICA E ANALISI TEMPORALE DELLE PARTITE 42
3.3.1 INDICATORI DI TENDENZA CENTRALE E MACRO-ANALISI DI LEGA 42
3.3.2 L' ANALISI DELLE SERIE STORICHE E DELLE FINESTRE TEMPORALI CRITICHE 47
3.4 DALLA MACRO-ANALISI ALLO SCOUTING TATTICO: LA STRUTTURA OPERATIVA DEL REPORT TATTICO 50
3.5 L'INTEGRAZIONE DEI LARGE LANGUAGE MODELS (LLM) E IL MODULO DI NATURAL LANGUAGE GENERATION (NLG) 54
3.5.1 IL PROBLEMA DELLA SINTESI E DELLA LATENZA COGNITIVA NEL CALCIO PROFESSIONISTICO 54
3.5.2 L'ARCHITETTURA DI NLG TRAMITE GEMINI PRO 54
CAPITOLO 4 60
4.1 INTRODUZIONE E VISIONE INGEGNERISTICA DEL SISTEMA 60
4.2 DIAGRAMMA CONCETTUALE DELLE CLASSI DI DOMINIO 61
4.3 MODELLAZIONE RELAZIONALE E SCHEMA DELLE CLASSI DI DOMINIO (3NF) 64
4.3.1 SCHEMA LOGICO ED ENTITÀ DI DOMINIO DEL SISTEMA 64
4.3.2 PRINCIPI DI NORMALIZZAZIONE E VANTAGGI ARCHITETTURALI (3NF) 66
4.3.3 IMPLEMENTAZIONE PRATICA DEI MODELLI DI DOMINIO 67
4.4 ARCHITETTURA DELL'INTERFACCIA UTENTE E ORGANIZZAZIONE DELLA DASHBOARD CENTRALE 69
4.4.1 LA SIDEBAR DI NAVIGAZIONE MODULARE E CONTROLLO DEGLI ACCESSI 69
4.4.2 LA BARRA DELLE AZIONI RAPIDE 71
4.5 L'ARCHITETTURA SAAS MULTI-TENANT E IL PANNELLO DI CONTROLLO CENTRALIZZATO (DJANGO ADMIN) 72
4.6 CRUSCOTTO DEI KPI MULTIDIMENSIONALI E GESTIONE FINANZIARIA 74
4.7 ANALISI DETTAGLIATA DEI MODULI OPERATIVI E DELLE TABELLE DI GESTIONE 75
4.7.1 MODULO "GESTIONE SQUADRE" 76
4.7.2 MODULO "CALENDARIO" 77

4.7.3 MODULO “GESTIONE CAMPI” & MODULO “PLANNING ORARI” 78
4.7.4 MODULO “MAGAZZINO KIT” 80
4.7.5 MODULO “GESTIONE ATLETI” 81
4.7.6 MODULO “AREA SCOUTING” 83
4.7.7 MODULO “STAFF E TECNICI” 84
4.8 LA SEZIONE DI SPORTS ANALYTICS 85
4.8.1 MODULO DATA ANALYSIS (GPS) 85
4.8.2 MODULO LAVAGNA TATTICA 87
4.8.3 MODULO REGISTRO INFORTUNI 89
4.8.4 MODULO TEST ATLETICI 92
4.8.5 MODULO MATCH ANALYSIS 95
4.9 ARCHITETTURA SOFTWARE E ORGANIZZAZIONE DEI COMPONENTI 98
4.9.1 SCOMPOSIZIONE MODULARE E ALBERATURA DEL PROGETTO 98
4.9.2 CICLO DI VITA DELLA RICHIESTA http E CUSTOM MIDDLEWARE DI LICENZA 99
4.10 LOGICA DI BUSINESS, FORM VALIDATION E TRANSAZIONI ATOMICHE (ACID) 101
CAPITOLO 5 102
5.1 L’USO DELLE PIATTAFORME E L’ABBATTIMENTO DELLA LATENZA DECISIONALE NEL CICLO SETTIMANALE 103
5.2 FEEDBACK QUALITATIVI E RISCONTRI OPERATIVI DELLO STAFF TECNICO 105
5.3 BILANCIO COMPLESSIVO DEI RISULTATI CONSEGUITI E VALIDAZIONE DELL’ARCHITETTURA. 107
5.4 LIMITI ATTUALI DELL’ANALISI DATI NEL CALCIO PROFESSIONISTICO 108
5.5 PROSPETTIVE ED EVOLUZIONI FUTURE: COMPUTER VISION E DEEP LEARNING NELLO SPORTS ANALYTICS 109
5.6 CONSIDERAZIONI CONCLUSIVE 110
BIBLIOGRAFIA 111

Introduzione

L'industria del calcio professionistico contemporaneo sta attraversando una fase di cambiamenti senza precedenti, paragonabile per portata e impatto alla rivoluzione che ha investito il settore finanziario e industriale con l'avvento dei Big Data.

Storicamente, la gestione di una squadra di calcio, dalla pianificazione settimanale della partita, alla campagna acquisti, passando per la gestione dei carichi di lavoro atletici, è stata guidata quasi esclusivamente da un approccio empirico, fondato sull'intuito visivo, sull'esperienza soggettiva dell'allenatore e su una tradizione orale tramandata di generazione in generazione all'interno degli staff tecnici e degli spogliatoi.

Sebbene questo paradigma abbia prodotto risultati straordinari nel corso dei decenni, esso soffre di intrinseci bias cognitivi: La percezione umana è infatti soggetta a distorsioni selettive, a un sovraccarico di informazioni visive in tempo reale e a una memoria episodica che tende a sovrappesare gli eventi recenti o emotivamente salienti rispetto alle tendenze macroscopiche di lungo periodo.

L'introduzione della **Sport Analytics** nel calcio non nasce con l'intento di sradicare la cultura del campo, bensì di affiancarla e potenziarla attraverso il rigore del metodo scientifico. Il passaggio a un modello decisionale di tipo Data-*Driven* risponde all'esigenza di ridurre l'area di incertezza che governa il rettangolo verde.

Il calcio, per sua natura, è uno sport complesso, non-lineare e caratterizzato da un bassissimo numero di eventi discreti determinanti, come ad esempio i gol, rispetto al tempo di gioco effettivo. Questa peculiarità statistica rende estremamente difficile isolare il merito tecnico o tattico di una prestazione basandosi unicamente sul risultato finale del tabellone.

L'analisi quantitativa interviene proprio qui: La partita viene scomposta in migliaia di micro-eventi (passaggi, duelli, pressioni, corse ad alta intensità, coordinate spaziali di tracciamento) e questi ultimi vengono aggregati in indicatori di performance oggettivi, capaci di misurare non solo ciò che è accaduto, ma soprattutto la probabilità che determinati eventi si ripetano in futuro.

La spinta verso la quantificazione scientifica non è guidata unicamente da motivazioni puramente tecnico-sportive, ma trova un fortissimo catalizzatore nella dimensione economica e industriale del calcio moderno.

I club professionistici odierni gestiscono capitali enormi, legati ai diritti televisivi, alle sponsorizzazioni e, soprattutto, alle elevate valutazioni di mercato dei calciatori e alla qualificazione alle competizioni europee. In un contesto in cui la retrocessione da una categoria superiore a una inferiore, o il mancato accesso a determinati obiettivi stagionali, comporta perdite di bilancio dell'ordine di decine di milioni di euro, il margine di errore nella gestione tecnica e strategica deve essere ridotto al minimo storico.

Investire nelle infrastrutture di dati, nell'acquisizione di feed di tracciamento avanzati e nello sviluppo di software proprietari non rappresenta più un lusso per pochi club d'élite, ma una necessità strategica di sopravvivenza e competitività. Le società che riescono a massimizzare l'efficienza nell'allocazione delle risorse attraverso i dati ottengono un vantaggio competitivo asimmetrico, sia nello scouting, individuando talenti sottovalutati sul mercato internazionale, sia nella preparazione tattica settimanale, confrontandosi con le cose giuste e sbagliate commesse nella settimana antecedente.

Dal punto di vista della Data Science, l'analisi del calcio presenta sfide metodologiche uniche e di gran lunga superiori rispetto ad altri sport. Discipline come il baseball o il basket sono caratterizzate da una forte interruzione del flusso di gioco, da un'alta frequenza di eventi di punteggio e da una chiara separazione delle situazioni di 1-contro-1, consentendo di isolare le performance individuali con modelli statistici relativamente lineari.

Il calcio, al contrario, è uno sport continuo, fluido, a flusso ininterrotto, dove ventidue giocatori interagiscono simultaneamente in uno spazio bidimensionale estremamente ampio e dinamico. La densità informativa è elevatissima, ma la validità statistica dei campioni è compromessa dalla ridotta numerosità temporale: Un'intera stagione calcistica si compone di un numero limitato di partite e i gol segnati si aggirano su medie ristrette.

La rarità degli eventi di punteggio genera una varianza stocastica enormemente spiccata. Di conseguenza, un data analyst che opera nel calcio non può limitarsi a leggere le statistiche descrittive superficiali, ma deve sviluppare modelli probabilistici avanzati.

Un esempio di questi modelli è rappresentato dai modelli di regressione logistica, nel caso degli expected goals. Un’altra tipologia di modelli è quella dei modelli spaziali.

Tali modelli devono essere in grado di estrarre il segnale, frutto di interesse, nascondendo il rumore di fondo.

All'interno di questo ecosistema complesso, il ruolo del Data Analyst in una società professionistica non si esaurisce dietro lo schermo di un computer a generare tabelle o grafici complessi.

Il valore scientifico di un'analisi è strettamente vincolato alla sua usabilità operativa: Un insight statistico, per quanto raffinato dal punto di vista matematico, risulta inutile se non viene tradotto in un linguaggio comprensibile, tempestivo e immediatamente azionabile per l'allenatore e i collaboratori di campo.

Lo staff tecnico vive in un regime di cronica compressione temporale e il Data Analyst deve agire come un traduttore bilingue capace di convertire la complessità degli algoritmi di Data Science e della teoria statistica in indicazioni tattiche concrete e rapide.

La validazione sul campo diventa così il banco di prova fondamentale di tutta la pipeline analitica: il riscontro visivo dell'allenatore e il confronto diretto con la realtà del rettangolo verde permettono di iterare i modelli, correggere le metriche e costruire un circolo virtuoso in cui la scienza dei dati convalida la metodologia di campo e, specularmente, la competenza calcistica valida la bontà delle assunzioni quantitative.

Il presente elaborato è strutturato nel seguente modo:

- Nel ***capitolo 1*** viene definito l'obiettivo generale della tesi, analizzando le problematiche aperte emerse sul campo e delineando formalmente i requisiti funzionali e non funzionali dei sistemi sviluppati.

- Nel ***capitolo 2*** vengono presentati i linguaggi di programmazione, i framework, le API, i database, i tool di Business Intelligence, le librerie di calcolo e gli ambienti di sviluppo impiegati nelle diverse fasi di realizzazione della tesi, fornendo una panoramica chiara delle risorse computazionali e software utilizzate.

- Nel ***capitolo 3*** viene approfondita la costruzione e la validazione del dataset proprietario (PREDICTO 2526) con lo staff dell'A.C. Perugia Calcio, formalizzando i modelli statistici di analisi e descrivendo l'integrazione con i Large Language Models (LLM) per la generazione automatizzata dei report tattici per l'allenatore.

- Nel ***capitolo 4*** viene trattata la progettazione architetturale e l'implementazione full-stack del sistema gestionale in Django, analizzando la struttura del database relazionale, la logica di dominio e la gestione dell'inserimento manuale e da file CSV.

- Nel ***capitolo 5*** viene discusso il riscontro pratico, accompagnato dai feedback ricevuti dall'allenatore e dallo staff tecnico, che hanno valutato l'efficacia operativa dei report predittivi e l'usabilità dello strumento gestionale nel contesto di un club professionistico. Inoltre, viene tracciato un bilancio complessivo sui risultati conseguiti, evidenziando i limiti attuali dell'analisi dati calcistica e proponendo possibili evoluzioni future.

CAPITOLO 1

1.1 OBIETTIVO GENERALE DELLA TESI

Il presente elaborato di tesi magistrale si pone l'obiettivo fondamentale di applicare i paradigmi, i modelli statistici e gli strumenti computazionali della Data Science all'interno dell'industria del calcio professionistico e dilettantistico, colmando il divario metodologico che storicamente ha separato la cultura empirica del campo dal rigore dell'analisi quantitativa.

Nel dettaglio, il lavoro si articola su due direttrici principali e complementari:

1. **La decodifica scientifica e l'analisi predittiva del contesto sportivo:** L'obiettivo primario di questa sezione è investigare, attraverso l'ingegneria del dato e la modellazione statistica, il settore della gestione tecnico-tattica sul campo, che per decenni è stato guidato quasi esclusivamente dall'esperienza pratica, dall'intuizione visiva e dalla tradizione orale degli staff tecnici. Piuttosto che da un oggettivo rigore scientifico. Sfruttando un dataset proprietario dal nome "**PREDICTO 2526.xlsx**", costruito e validato iterativamente sul campo durante l'esperienza all'interno dell'A.C. Perugia Calcio, la tesi mira a estrarre pattern tattici nascosti, a studiare l'andamento macroscopico del campionato e a strutturare report di scouting predittivo per la preparazione della gara contro l'avversario di turno. A supporto di questa pipeline, viene introdotto l'utilizzo di Large Language Models (LLM) come livello di Natural Language Generation (NLG), con lo scopo di tradurre matrici numeriche complesse e tabelle statistiche in indicazioni testuali immediate e prescrittive, riducendo drasticamente il carico cognitivo e la latenza decisionale dell'allenatore e del rispettivo staff tecnico, senza però andare ad intaccare l'importanza delle figure di questi ultimi.

2. **L'ingegnerizzazione di un sistema gestionale full-stack a supporto delle società sportive:**
   Accanto alla componente di analisi tattica, la tesi affronta il problema della digitalizzazione e della centralizzazione operativa delle società calcistiche a livello di un'unica piattaforma (sia a livello di ASD che di club professionistici).
   L'obiettivo tecnico è la progettazione e lo sviluppo di un'applicazione web robusta, scalabile e basata sul framework Django.
   Tale applicazione ha preso come nome "**gestionale_asd_calcio"**.
   Il gestionale risponde all'esigenza di superare la frammentazione informativa e la disomogeneità dei formati di input (gestione manuale e flussi CSV). Attraverso un'architettura Model-View-Template (MVT) e un database relazionale fortemente normalizzato, il sistema centralizza la segreteria classica (anagrafiche di atleti e collaboratori, tesseramenti, scadenze mediche), la logistica (magazzino kit e abbigliamento sportivo), la pianificazione agonistica (calendari gare e convocazioni) e moduli di visualizzazione rapida dei KPI, fornendo un'infrastruttura software solida e pronta per l'uso sul campo.

1.2 PROBLEMATICHE APERTE NEL DOMINIO DI RIFERIMENTO

L'integrazione della scienza dei dati e delle soluzioni digitali all'interno di un club calcistico professionistico (es: A.C. Perugia Calcio) e nel più ampio panorama delle associazioni sportive dilettantistiche fa emergere criticità strutturali, operative e metodologiche che questo lavoro si propone di affrontare:

- **La compressione temporale e il sovraccarico cognitivo dello staff tecnico:** Nel calcio professionistico, la frequenza serrata degli impegni agonistici settimanali impone tempi di reazione estremamente ridotti. Gli allenatori e i collaboratori tecnici operano sotto una cronica pressione temporale, disponendo di finestre ristrette per analizzare la mole di dati grezzi prodotta durante le gare. La problematica aperta risiede nella difficoltà di sintetizzare matrici numeriche complesse in indicazioni tattiche immediate, prive di rumore e pronte per essere assimilate dai calciatori prima delle sedute video o di campo, che ugualmente sono caratterizzate da una durata piuttosto ridotta.

- **La qualità, il rumore e l'eterogeneità del dato sportivo:** I dati calcistici sono molto rumorosi e complessi da modellare, a causa della natura continua del gioco ed essendo caratterizzati da una bassa numerosità degli eventi che si traducono in gol. Alla seguente problematica si aggiunge, nel contesto delle di categorie minori, l'assenza di sistemi di tracciamento ottico o sensori di prima fascia costosi, rendendo indispensabile la gestione di flussi informativi disomogenei, basati su inserimenti manuali o tracciati CSV strutturati da operatori umani.

- **La frammentazione e la disarticolazione gestionale:** Molte realtà sportive soffrono della mancanza di un ecosistema software centralizzato. La gestione amministrativa e di segreteria, la logistica del magazzino) e il monitoraggio delle gare, avvengono spesso tramite strumenti disarticolati e non condivisi. Questa frammentazione compromette l'integrità dei dati, rende onerosa la tracciabilità e impedisce una visione d'insieme della salute organizzativa del club.

1.3 ANALISI E DEFINIZIONE DEI REQUISITI

Per rispondere in modo strutturato alle problematiche individuate, l'architettura complessiva sviluppata in questo lavoro si suddivide in due macro-componenti che fra loro risultano essere complementari: La pipeline di analisi tattica/predittiva e il sistema gestionale full-stack. Di seguito vengono formalizzati i relativi requisiti funzionali e non funzionali.

Si specifica che i requisiti funzionali verranno presentati con la sigla *RF*, mentre i requisiti non funzionali verranno presentati con la sigla *RNF*.

1.3.1 REQUISITI PER LA PIPELINE DI ANALISI TATTICA E REPORTISTICA (Il Caso A.C. Perugia Calcio)

- ***REQUISITI FUNZIONALI***

    - *RF1 - Ingestion, normalizzazione e validazione del dataset:* Il sistema deve acquisire, strutturare e normalizzare flussi di dati eterogenei, provenienti da feed di match-stats e dataset storici di intero campionato, per popolare il database di lavoro (PREDICTO 2526.xlsx). Ciascuna metrica di prestazione, come ad esempio il possesso palla medio o i tiri totali di una relativa squadra, deve essere validata iterativamente tramite confronto diretto con lo staff dell'A.C. Perugia Calcio.

    - *RF2 - Analisi temporale e scouting dell'avversario:* Il sistema deve calcolare metriche comparative di rendimento tra la squadra e l'avversario di turno, analizzando il rendimento storico relativo al giocare in casa o in trasferta, le statistiche delle ultime cinque giornate disputate e mappando le finestre temporali critiche, suddivise in sei intervalli da 15 minuti, per individuare i momenti di blackout difensivo/offensivo e la vulnerabilità strutturale su determinati

fondamentali di gioco, come ad esempio i gol subiti da palle inattive o da cross tagliati.

- *RF3 - Generazione testuale automatica tramite LLM (NLG):* Il sistema deve integrare un modulo basato su Large Language Models, che riceva in input le matrici numeriche elaborate e le tabelle comparative, restituendo sezioni testuali discorsive, strutturate e prescrittive, così da tradurre il dato quantitativo in direttive tattiche immediatamente fruibili dall'allenatore.

- ***REQUISITI NON FUNZIONALI***

  - *RNF1 - Accuratezza e tracciabilità metodologica*: Le metriche derivate devono mantenere coerenza matematica con i dati storici di campionato, garantendo tracciabilità e rigore scientifico.

  - *RNF2 - Usabilità e rapidità di delivery:* Il report finale deve essere esportato in un formato tipografico agile, ottimizzato per una consultazione visiva e testuale immediata, compatibile con i ritmi di lavoro ristretti dello staff tecnico prima della seduta video.

1.3.2 REQUISITI PER IL SISTEMA GESTIONALE FULL-STACK (gestionale_asd_calcio)

- ***REQUISITI FUNZIONALI***

  - *RF4 Sistema di autenticazione e profilazione:* Il sistema deve fornire funzionalità di autenticazione e profilazione degli utenti, permettendo di associare ciascun utente alla propria specifica società sportiva di riferimento.

- *RF5 - Gestione anagrafica, segreteria e tracciamento sanitario:* Il sistema deve gestire le entità anagrafiche degli atleti e dei collaboratori, monitorando lo stato dei tesseramenti e le scadenze medico-sportive tramite calcoli temporali automatizzati, come ad esempio i giorni rimanenti alla scadenza delle visite mediche.

- *RF6 - Gestione logistica del magazzino kit e abbigliamento:* Il sistema deve tracciare la disponibilità degli articoli sportivi provenienti dal magazzino, vincolati a tipologie d'articolo predefinite e taglie standardizzate, associando le giacenze alle singole società.

- *RF7 - Pianificazione agonistica e gestione convocazioni:* Il sistema deve permettere la pianificazione delle gare, distinguendo la tipologia tra amichevoli e gare di campionato o tornei, registrando la squadra avversaria, la data, l'ora e il campo, e gestendo le convocazioni con l'entità relativa agli atleti.

- *RF8 - Inserimento manuale, parsing CSV e data visualization:* Il sistema deve fornire form web e parser dedicati per normalizzare l'inserimento manuale e l'importazione di file CSV da parte di utenti non tecnici, alimentando dashboard interattive per la visualizzazione rapida dei KPI di squadra e individuali, come ad esempio le presenze in campo, l'entità degli infortuni oppure i trend degli atleti.

- ***REQUISITI NON FUNZIONALI***

- *RNF3 - Robustezza architetturale (pattern Model-View-Template):* Il software deve essere sviluppato utilizzando il framework Django, sfruttando la netta separazione dei compiti dell'architettura MVT, l'efficienza del routing e le robustezze del pannello di amministrazione nativo.

- *RNF4 - Integrità referenziale e vincoli di dominio (ORM):* L'Object-Relational Mapping (ORM) di Django e il database sottostante (SQLite/MySQL) devono applicare rigorosi vincoli di integrità referenziale per prevenire corruzioni di dati.

- *RNF5 - Scalabilità ed estendibilità modulare:* La struttura del codice e l'albero delle viste, visibili nello stack di sviluppo con moduli dedicati alle varie sezioni, come ad esempio quella degli infortuni o della lavagna tattica, devono essere altamente modulari, consentendo l'innesto futuro di nuove metriche di performance atletica, ma anche di nuovi moduli per l'ampliamento delle tematiche coperte.

- RNF6 - *Sicurezza e compartimentazione dei dati*: Il sistema deve garantire l'isolamento, la sicurezza e la rigorosa compartimentazione dei dati tra differenti società sportive, che siano ASD o club professionistici, prevenendo qualsiasi accesso o visualizzazione non autorizzata tra enti distinti.

CAPITOLO 2

L'architettura software e computazionale sviluppata per il presente elaborato si basa su un ecosistema tecnologico eterogeneo ma fortemente integrato, progettato per soddisfare i requisiti di robustezza, scalabilità e rigore analitico propri della Data Science e dello sviluppo di sistemi full-stack. In questo capitolo vengono analizzati in modo approfondito i linguaggi di programmazione, i framework, le interfacce di programmazione (API), i modelli linguistici, i database e gli ambienti di sviluppo impiegati nelle diverse fasi della pipeline, evidenziando il ruolo e il valore aggiunto di ciascuno strumento.

2.1 LINGUAGGI DI PROGRAMMAZIONE & CORE TECHNOLOGIES

2.1.1 PYTHON (V3.X)



Figura 2.1

Python (*Figura 2.1*) rappresenta la spina dorsale dell'intera infrastruttura computazionale sviluppata in questa tesi, fungendo da linguaggio principale sia per la pipeline di data engineering e manipolazione statistica, sia per la logica di backend del sistema gestionale. [1]

La scelta di Python è motivata da molteplici fattori chiave in ambito accademico e industriale.

Python è caratterizzato da un ricco ecosistema di librerie per la Data Science. La disponibilità nativa o tramite package manager di librerie altamente ottimizzate in C (quali Pandas per la manipolazione di dataframe tabulari e NumPy per il calcolo vettorializzato) ha permesso di gestire con estrema efficienza la pulizia, il filtraggio e l'aggregazione del dataset proprietario di campionato.

Inoltre, Python è noto ai più per la sua flessibilità e la leggibilità della sintassi. La vicinanza del linguaggio al linguaggio naturale ha facilitato lo sviluppo rapido del backend basato su Django e l'integrazione di script complessi per il parsing di file CSV e l'interfacciamento con servizi esterni.

Un altro elemento a favore del linguaggio di programmazione scelto è dato dal supporto alla programmazione orientata agli oggetti (OOP). L'adozione di paradigmi a oggetti ha garantito una netta separazione delle responsabilità nella strutturazione delle classi di dominio del gestionale.

2.1.2 R STUDIO



Figura 2.2

Accanto a Python, il linguaggio **R** (*Figura 2.2*) è stato integrato all'interno della metodologia di ricerca per l'esecuzione di analisi statistiche avanzate, modellazione di dati sportivi e validazione incrociata dei test d'ipotesi applicati alle metriche di prestazione calcistica.[2]

R eccelle nativamente nella gestione di modelli statistici complessi e nella rappresentazione grafica esplorativa, grazie a librerie come ggplot2, offrendo un riscontro metodologico complementare rispetto all'ambiente Python e garantendo la robustezza formale dei risultati statistici estratti dal campionato.

2.1.3 HTML, CSS e JAVASCRIPT

Per la componente di interfaccia utente del sistema gestionale, sono stati impiegati i linguaggi fondamentali del Web:



Figura 2.3

- **HTML (HyperText Markup Language**) *(Figura 2.3)***:** Utilizzato per la strutturazione semantica dei template di visualizzazione gestiti dal framework Django (pattern Model-View-Template). [3]



Figura 2.4

- **CSS (Cascading Style Sheets)** *(Figura 2.4)***:** Impiegato per la formattazione grafica, il design responsivo e l'implementazione di layout puliti, garantendo un'esperienza utente (UX) coerente e funzionale per operatori non necessariamente tecnici, come ad esempio impiegati di segreteria, addetto allo staff e dirigenti. [4]



Figura 2.5

- **JavaScript** *(Figura 2.5)***:** Adoperato per introdurre comportamenti dinamici e interattivi lato client, consentendo la gestione di eventi asincroni e la validazione preliminare dei form prima dell'invio delle richieste HTTP al server. [5]

2.2 FRAMEWORK DI SVILUPPO WEB E INTERFACCE API

2.2.1 DJANGO



Figura 2.6

Il framework **Django** *(Figura 2.6)* (scritto in Python) è stato scelto per lo sviluppo dell'intero sistema gestionale full-stack data la sua natura completa e versatile che fornisce nativamente tutto il necessario per la costruzione di applicazioni web robuste e sicure [6].

Tra le caratteristiche architetturali che hanno guidato la scelta, non si può non citare il **pattern Model-View-Template (MVT)**, ovvero una variante del classico pattern MVC, che garantisce una separazione rigorosa tra la definizione dello schema dati (Model), la logica di business e di controllo (View) e la presentazione visiva (Template).

Un'altra caratteristica degna di nota è data dall'**Object-Relational Mapping (ORM)** nativo di Django, che ha permesso di mappare le entità del dominio sportivo, quali il modulo società, il modulo atleti, il modulo gare ..., direttamente in classi Python, traducendo automaticamente le operazioni in query SQL ottimizzate e gestendo con sicurezza i vincoli relazionali complessi.

Importanti sono anche le protezioni integrate ed automatiche che Django offre contro le principali vulnerabilità web OWASP Top 10, quali Cross-Site Request Forgery (CSRF), SQL injection, Clickjacking e gestione non sicura delle password tramite hashing crittografato.

Infine, Django dispone di un admin panel, un pannello di amministrazione autogenerato che ha permesso di accelerare notevolmente il testing e la gestione amministrativa dei dati anagrafici e di magazzino durante le fasi di sviluppo.

2.2.2 INTERFACCE API (Application Programming Interfaces)

Nel contesto di questa tesi, le **API** assumono una duplice valenza architetturale.

All'interno del progetto vengono utilizzate **API interne di sistema** per la gestione delle richieste asincrone, tramite chiamate AJAX/Fetch, e per la strutturazione dei flussi di comunicazione tra i componenti client e server all'interno del gestionale.

Vengono inoltre utilizzate **API di servizi esterni e Large Language Models**, ovvero interfacce di programmazione basate su protocolli REST e SDK dedicati, come ad esempio Google GenAI SDK per l'interrogazione di Gemini Pro, utilizzate per inviare prompt ingegnerizzati contenenti metriche numeriche e ricevere in risposta dei testi descrittivi e riepilogativi da inserire nei report tattici.

2.3 INTEGRAZIONE DI MODELLI DI LINGUAGGIO (LLM)

2.3.1 GEMINI PRO (GOOGLE GenAI)



Figura 2.7

Nel panorama della Data Science applicata allo sport, uno dei limiti storici è rappresentato dal divario tra la complessità delle matrici numeriche e la necessità di una fruizione immediata da parte degli operatori di campo. Per superare questa barriera, all'interno della pipeline di reportistica tattica è stata integrata la tecnologia dei Large Language Models (LLM), sfruttando in particolare **Gemini Pro** *(Figura 2.7)* [7].

Il modello non è stato impiegato come semplice chatbot, bensì come motore di generazione di testo condizionato. A valle dell'elaborazione deterministica dei dati grezzi e delle tabelle comparative all'interno del dataset di campionato, gli insight quantitativi sono stati incapsulati in prompt ingegnerizzati e passati programmaticamente all'API di Gemini Pro.

Una volta effettuato questo passaggio, il Large Language Model ha assolto il compito di tradurre i dati strutturati in sezioni testuali discorsive, rigorose e immediatamente prescrittive, riducendo drasticamente il carico cognitivo dell'allenatore e del suo staff tecnico e velocizzando il processo decisionale.

2.4 BUSINESS INTELLIGENCE, VISUALIZZAZIONE E FOGLI DI CALCOLO

2.4.1 POWER BI



Figura 2.8

Nel panorama della Sport Analytics moderna, la capacità di comunicare l'insight analitico in modo rapido ed efficace riveste la stessa importanza della bontà del modello statistico sottostante. Infatti, la mancata comprensione dei report e di quanto scritto in questi ultimi, danneggerebbe il lavoro stesso, rendendolo inutile.

Per soddisfare questa esigenza, all'interno della pipeline di monitoraggio e presentazione dei dati è stato integrato **Power BI** *(Figura 2.8)* [8].

Il suo utilizzo è risultato fondamentale in quanto Power BI è stato impiegato per connettere fonti di dati eterogenee, strutturare modelli relazionali a stella e calcolare misure dinamiche avanzate tramite il linguaggio DAX (Data Analysis Expressions).

Lo strumento ha permesso di tradurre tabelle numeriche complesse in dashboard visive interattive, consentendo all'allenatore e ai preparatori di esplorare i KPI di prestazione, sia collettivi che individuali, filtrando istantaneamente i dati per condizione di campo, ovvero distinguendo le partite in

casa ed in trasferta, fasce temporali di gara e metriche di efficienza offensiva e difensiva. Questa componente visiva ha agito da ponte fondamentale tra la complessità della Data Science e la fruibilità operativa sul campo.

2.4.2 MICROSOFT EXCEL



Figura 2.9

Nelle fasi iniziali di progettazione, raccolta e prototipazione della pipeline di analisi, **Microsoft Excel** *(Figura 2.9)* ha ricoperto un ruolo operativo insostituibile [9].

Excel è stato utilizzato come ambiente primario per l'assemblaggio, la strutturazione tabulare e la prima normalizzazione del dataset di campionato. La griglia flessibile del foglio di calcolo ha facilitato l'identificazione di valori anomali, errori di battitura e inconsistenze nei flussi di dati grezzi.

Excel ha inoltre svolto la funzione di strumento di interfaccia immediata durante le sessioni di confronto con lo staff tecnico dell'A.C. Perugia Calcio. La possibilità di filtrare e visualizzare istantaneamente i dati ha permesso di eseguire una validazione iterativa basata sul feedback di dominio, prima di procedere alla migrazione definitiva dei dati verso ambienti relazionali strutturati e script di automazione in Python.

2.5 GESTIONE DI DATI, BASI DI DATI & LIBRERIE DI CALCOLO

2.5.1 MYSQL e MYSQLite



Figura 2.10



Figura2.11

La persistenza strutturata dei dati, cruciale sia per la tenuta dello storico analitico sia per il corretto funzionamento del sistema gestionale, è stata affidata a database relazionali (RDBMS) scelti in base alle specifiche esigenze di sviluppo e deployment:

- **MySQL** *(Figura 2.10)***:** MySQL è stato selezionato come sistema di database relazionale di riferimento per l'ambiente di produzione [10].

MySQL garantisce un'elevata robustezza nella gestione della concorrenza transazionale, il pieno supporto ai vincoli di integrità referenziale (chiavi esterne, politiche di cancellazione a cascata) e la scalabilità necessaria per gestire i volumi di dati anagrafici, i calendari delle gare, i log di magazzino e i flussi di dati importati tramite file CSV.

- **SQLite** *(Figura 2.11)*: Adoperato durante le fasi iniziali di sviluppo locale e prototipazione rapida [11]. Grazie alla sua natura di database basato su file zero-configuration e integrato nativamente nel framework Django, SQLite ha permesso di accelerare la scrittura del codice, la validazione delle migrazioni e la portabilità dell'applicazione senza l'overhead di configurazione di un server di database dedicato.

2.5.2 PANDAS e NUMPY



Figura 2.12



Figura 2.13

Per l'esecuzione di tutte le operazioni di manipolazione, pulizia, trasformazione e analisi statistica all'interno degli script Python, sono state impiegate le librerie cardine dell'ecosistema scientifico:

- **Pandas** *(Figura 2.12)***:** Utilizzato per la gestione di strutture dati tabulari bidimensionali eterogenee, ovvero i DataFrame[12]. Pandas ha permesso di eseguire operazioni complesse di data wrangling, tra cui il filtraggio condizionale, il merge di dataset provenienti da fonti differenti, la gestione dei valori mancanti e l'aggregazione temporale dei dati per singola giornata di campionato.

- **NumPy** *(Figura 2.13)***:** Impiegato per il supporto al calcolo numerico ad alte prestazioni, grazie alla gestione ottimizzata di array multidimensionali omogenei e di funzioni matematiche vettorizzate, riducendo drasticamente i tempi di esecuzione computazionale rispetto ai cicli iterativi nativi di Python [13].

2.6 AMBIENTI DI SVILUPPO, CONTROLLO DI VERSIONE e TRACCIABILITÀ

2.6.1 VISUAL STUDIO CODE (VS CODE)



Figura 2.14

Per la stesura, il debug e la gestione quotidiana del codice sorgente dell'intero progetto è stato adottato **Visual Studio Code** *(Figura 2.14)*, apprezzato per la sua architettura leggera, altamente modulare ed estendibile [14].

L'IDE è stato configurato tramite l'installazione di estensioni e moduli specifici per abilitare il supporto avanzato a Python, Django, HTML/CSS/JavaScript e Git.

VS Code ha fornito funzionalità essenziali per la qualità del software, quali l'autocompletamento intelligente basato su IntelliSense, il controllo sintattico on-the-fly, i terminali integrati per l'esecuzione dei comandi di gestione e migrazione dei database, e un debugger interattivo ottimizzato per il flusso di lavoro rapido tipico dello sviluppo web e della Data Science.

2.6.2 GIT e GITHUB



Figura 2.15



Figura 2.16

La gestione del codice sorgente, l'evoluzione architetturale del software e la tracciabilità delle modifiche sono state affidate a **Git** *(Figura 2.15)*, integrato con la piattaforma di hosting cloud **GitHub** *(Figura 2.16)*.

L'adozione di un flusso di lavoro basato su rami di sviluppo isolati ha permesso di implementare nuove funzionalità, sia nella pipeline del dataset proprietario relativo al campionato che nei moduli del gestionale Django, senza compromettere la stabilità del codice principale.

L'utilizzo di commit atomici e descrittivi ha garantito la completa tracciabilità cronologica delle modifiche, offrendo un sistema di backup sicuro e assicurando la piena riproducibilità scientifica e ingegneristica dell'intero progetto di tesi.

CAPITOLO 3

3.1 INTRODUZIONE E ARCHITETTURA DELLA PIPELINE DEI DATI

Nel dominio della Sport Analytics applicata al calcio professionistico, il valore di un modello di Data Science non risiede esclusivamente nella complessità degli algoritmi predittivi impiegati o nella raffinatezza formale degli stimatori statistici, bensì nella robustezza e nella scalabilità dell'intera *Data Pipeline*.

Quest'ultima si configura come un flusso ingegneristico strutturato e continuo, progettato per connettere la raccolta empirica dei dati grezzi di campo con la restituzione tempestiva di indicazioni strategiche direttamente fruibili dallo staff tecnico.

A differenza dei contesti aziendali tradizionali, tipicamente caratterizzati da flussi informativi stabili, asincroni e orientati a finestre decisionali di medio-lungo termine, il calcio professionistico opera in un regime di rigida **compressione temporale e operativa**: la frequenza settimanale, e talvolta infrasettimanale, degli impegni agonistici impone finestre di analisi estremamente ristrette, spesso inferiori alle 72 ore che separano il triplice fischio di una gara ufficiale dalla prima seduta di preparazione tattica per l'incontro successivo. In questo scenario ad alta intensità, un ritardo o un'inconsistenza nella catena di elaborazione del dato vanificherebbe l'efficacia dell'intero modello analitico.

Per rispondere a tale criticità strutturale e garantire un supporto decisionale rigoroso e puntuale, è stata progettata e implementata una pipeline *end-to-end* completamente automatizzata, la cui architettura complessiva è sintetizzata nello schema a blocchi sottostante *(Figura 3.1)*.

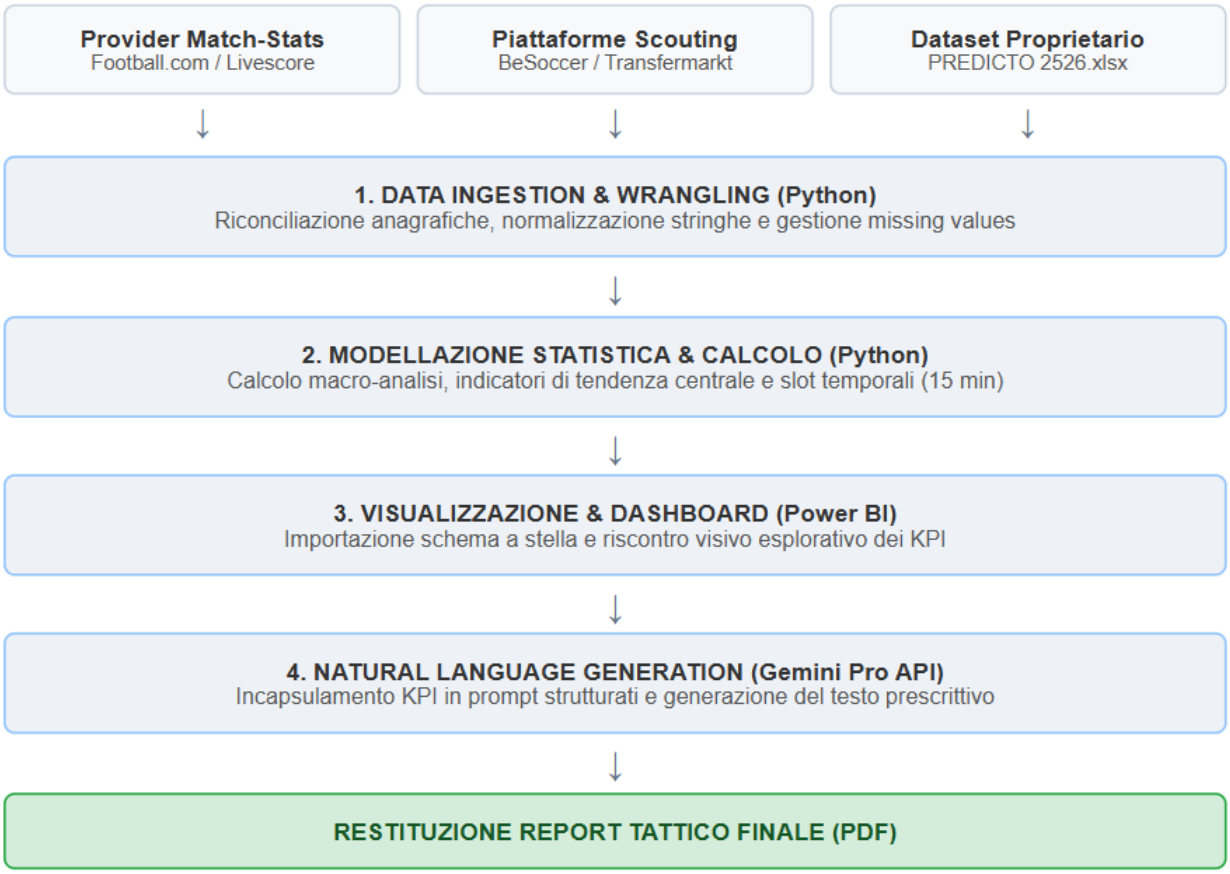


Figura 3.1

La prima fase della pipeline è rappresentata dal **data engineering e ingestion**. L'intero processo parte dall'acquisizione, sincronizzazione e pulizia dei flussi di dati grezzi provenienti da fonti eterogenee e distribuite, tra le quali figurano provider di match-stats in tempo reale, piattaforme di scouting avanzato ed il dataset storico di campionato. Questa fase abilita la trasformazione di informazioni non strutturate o semi-strutturate in matrici relazionali pronte per il calcolo.

Lo step successivo riguarda la **modellazione statistica ed il calcolo** e tale operazione avviene grazie all'utilizzo del linguaggio Python. È in questa fase che avviene l'aggregazione avanzata dei KPI di squadra e di reparto, l'analisi delle serie storiche temporali suddivise in finestre di 15 minuti e il calcolo delle metriche di prestazione strutturale, come le percentuali di possesso palla, gli indici di aggressività disciplinare e le distribuzioni cronologiche delle reti realizzate e subite.

La terza fase, svolge il ruolo di ponte tra i risultati numerici ottenuti con la successiva realizzazione del report descrittivo. Tale operazione avviene con la **visualizzazione tramite Power BI**. Il modello relazionale viene importato all'interno dell'ambiente di Business Intelligence per consentire un'esplorazione visiva rapida e un riscontro interattivo dei KPI di campionato.

La fase finale della pipeline riguarda la **Natural Language Generation (NLG)** tramite Large Language Models. In questo momento avviene l'integrazione di modelli di linguaggio avanzati, in questo caso Gemini Pro, all'interno della pipeline. Le matrici numeriche complesse e le tabelle comparative vengono così tradotte in sezioni testuali discorsive, rigorose e prescrittive, di facile comprensione per lo staff tecnico. Queste sezioni vanno a costituire il report tattico in formato PDF.

Questa architettura modulare non solo assicura rigore scientifico e riproducibilità all'analisi, ma azzera i tempi di decisione, offrendo allo staff tecnico un collegamento immediato tra la complessità della Data Science e la concretezza del campo da gioco.

3.2 INGEGNERIA DEL DATO: FONTI ETEROGENEE, STRUTTURA E VALIDAZIONE DEL DATASET.

3.2.1. L'ECOSISTEMA DELLE FONTI E I CANALI DI INGESTIONE

La costruzione della pipeline analitica si basa su un processo di d*ata ingestion* multicanale, progettato per superare i limiti dei singoli provider e aggregare informazioni di match analysis, profili individuali di scouting, valori di mercato e dati storici di campionato. Nello specifico, l'infrastruttura informativa attinge da quattro macro-categorie di fonti esterne e interne, integrate all'interno del database di lavoro:

1. *Provider di risultati e match-stats in tempo reale (Football.com [17] / Livescore [18]):* Utilizzati per l'estrazione e l'aggiornamento costante dei calendari ufficiali, dei tabellini di gara e delle metriche aggregate per giornata relative al girone di riferimento, che in questo è il girone B della Serie C. Questi flussi forniscono la base per il tracciamento dei volumi di squadra, ovvero possesso palla, tiri totali, tiri in porta, calci d'angolo e falli. I dati ricavati sono necessari per popolare le matrici di rendimento casalingo e in trasferta.

2. *Piattaforme di scouting avanzato e profili individuali (BeSoccer [19] & Transfermarkt [20]):* Integrate per arricchire il dataset con informazioni sui singoli atleti, dati anagrafici, storici di rendimento individuale, metriche di impatto realizzativo e valutazioni economiche. L'estrazione di questi dati permette di passare da un'analisi puramente macroscopica di squadra a un'indagine microanalitica focalizzata sulle individualità determinanti dell'avversario.

3. *Ambienti di elaborazione e clustering dedicati*: Utilizzati come ambienti di sviluppo e testing per script Python mirati alla classificazione e al

raggruppamento delle squadre in base a similitudini tattiche e comportamentali. Tali moduli consentono di eseguire algoritmi di apprendimento non supervisionato, come l'algoritmo K-Means oppure il clustering gerarchico, per identificare cluster di squadre con caratteristiche difensive o offensive analoghe, offrendo allo staff un livello di categorizzazione avanzata oltre la semplice classifica di campionato.

4. *Il dataset proprietario di sintesi (**PREDICTO 2526.xlsx**):* Il raccoglitore centrale e strutturato in cui confluiscono tutte le informazioni elaborate. Il file organizza i dati in molteplici fogli di lavoro logici:

    - **CALENDARIO CON RISULTATI** *(Figura 3.2)*: Storico degli incontri e distinzione topologica casa/trasferta.

MATCH_ID	GIORNATA	SQUADRA CASA	SQUADRA OSPITE	RISULTATO
1	1	SAMBENEDETTESE	BRA	1-0
2	1	PINETO	VIS PESARO	1-1
3	1	AREZZO	FORLI	1-0

Figura 3.2

    - **ELENCO GOL FATTI** ed **ELENCO GOL SUBITI** *(Figura 3.3; Figura 3.4)*: Granularità dei gol realizzati e subiti, mappati per minuto esatto e modalità realizzativa (tiro su cross, colpo di testa, tiri da fuori, calci piazzati).

MATCH_ID	GIORNATA	SQUADRA	GIOCATORE	MODALITA GOL FATTO	MINUTO	CASA(0) / TRASFERTA(1)
1	1	SAMBENEDETTESE	KONATE A.	TIRO SU CROSS	35	0
2	1	PINETO	JALLOW S.	TIRO SU CROSS	8	0
2	1	VIS PESARO	GERMINARIO G.	TIRO SU CROSS	17	1
3	1	AREZZO	TAVERNELLI C.	TIRO DA FUORI AREA	50	0

Figura 3.3

MATCH_ID	GIORNATA	SQUADRA	MODALITA GOL SUBITO	MINUTO GOL SUBITO	CASA(0) / TRASFERTA(1)
1	1	BRA	TIRO SU CROSS	35	1
2	1	PINETO	TIRO SU CROSS	17	0
2	1	VIS PESARO	TIRO SU CROSS	8	1
3	1	FORLI	TIRO DA FUORI AREA	50	1

Figura 3.4

- **ELENCO CARTELLINI** *(Figura 3.5)*: Tracciamento disciplinare di gialli e rossi per la stima dell'aggressività.

MATCH_ID	GIORNATA	SQUADRA	CARTELLINI GIALLI	CARTELLINI ROSSI	CASA(0) / TRASFERTA(1)
1	1	SAMBENEDETTESE	2	0	0
1	1	BRA	2	0	1
2	1	PINETO	2	0	0
2	1	VIS PESARO	5	0	1
3	1	AREZZO	2	0	0
3	1	FORLI	4	0	1

Figura 3.5

- **STATISTICHE COMPLETE x GIORNATA** *(Figura 3.6)*: Matrice multidimensionale dei KPI prestazionali per singolo match.

MATCH_ID	GIORNATA	SQUADRA	POSSESSO PALLA	TIRI	TIRI IN PORTA	CALCI D'ANGOLO	CASA(0) / TRASFERTA(1)	FALLI	FUORIGIOCO
1	1	SAMBENEDETTESE	52	12	3	6	0	18	2
1	1	BRA	48	11	2	7	1	15	0
2	1	PINETO	51	7	2	5	0	20	1
2	1	VIS PESARO	49	7	5	6	1	28	0
3	1	AREZZO	61	14	7	6	0	16	0
3	1	FORLI	39	9	3	4	1	16	2

Figura 3.6

3.2.2 NORMALIZZAZIONE, SINCRONIZZAZIONE E DATA WRANGLING TRAMITE SCRIPT PYTHON

L'eterogeneità delle fonti descritte impone una fase rigorosa di *Data wrangling* e pulizia implementata tramite script Python, attraverso le librerie *Pandas* e *NumPy*.

Tale operazione risulta essere particolarmente importante per molteplici motivazioni, quali la **riconciliazione degli identificativi e delle anagrafiche**. Sono presenti vari casi da tenere in considerazione sia nei nomi delle squadre (La Juventus Next Gen assume in alcuni casi il nome di Juventus, oppure il nome di Juventus II), sia nei nomi dei giocatori, specialmente nel caso di doppio nome e doppio cognome. I nomi delle squadre e dei giocatori provenienti da piattaforme differenti vengono normalizzati attraverso dizionari di mappatura univoci per prevenire errori di join relazionale all'interno del dataset.

Risulta di fondamentale importanza anche **la gestione dei flussi asincroni e dei missing values.** Tale operazione avviene con la sincronizzazione dei dati estratti dai match live con lo storico consolidato, applicando logiche di imputazione per eventuali metriche mancanti e verificando la coerenza temporale delle giornate di campionato.

Le librerie utilizzate risultano infine fondamentali per la creazione dei dataset tabulari, facilmente interrogabili sia dai modelli statistici descrittivi sia dai moduli di generazione automatica dei report tattici in PDF.

3.2.3 LA VALIDAZIONE DI DOMINIO CON LO STAFF TECNICO

Anche a fronte di un'ingente mole di dati estratti da provider esterni, il modello non prescinde dal controllo umano, dato che la mancanza di comprensione andrebbe a rendere vano tutto il lavoro svolto.

Il processo di validazione congiunta con l'allenatore dell'A.C. Perugia Calcio, e con il rispettivo staff, garantisce che le metriche complesse e i cluster elaborati rispecchino fedelmente le dinamiche del campo, massimizzando la fiducia dello staff operativo nei report forniti.

3.2.4 MODELLAZIONE RELAZIONALE ED INTEGRAZIONE IN POWER BI

Per consentire un'esplorazione visiva e un riscontro interattivo dei parametri di campionato, l'intera struttura informativa preventivamente estratta e processata tramite gli script Python è stata importata all'interno dell'ambiente di Business Intelligence di Power BI. *(Figura 3.7)*.

All'interno dell'architettura della pipeline, il ruolo di Power BI è circoscritto esclusivamente alla visualizzazione grafica dei KPI e alla consultazione interattiva del modello relazionale, fungendo da livello visivo di supporto per la verifica immediata delle metriche di prestazione.

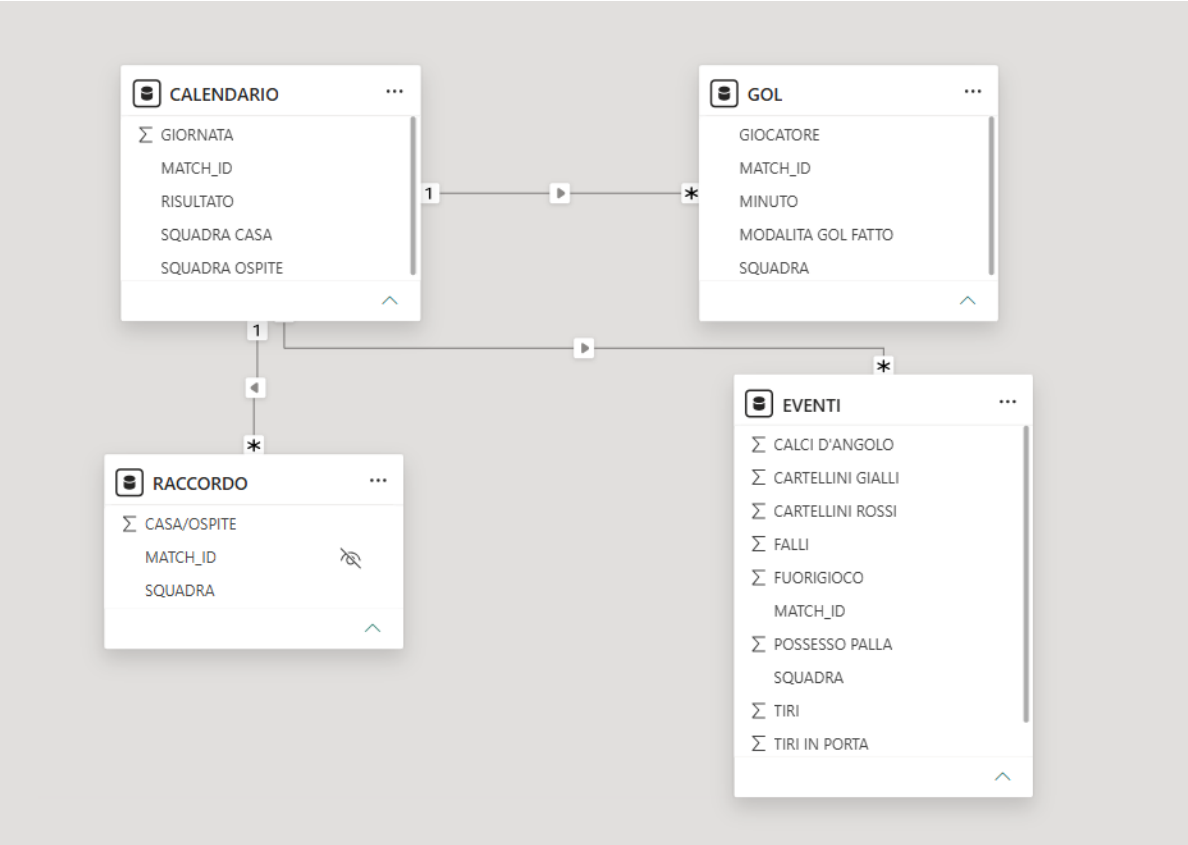


Figura 3.7

L'architettura dei dati è stata organizzata in uno schema relazionale ottimizzato per la gestione degli eventi sportivi:

- **La tabella CALENDARIO:** Funge da entità primaria di riferimento per la stagione, censendo l'identificativo univoco dell'incontro (MATCH_ID), la giornata di campionato (GIORNATA), le squadre contrapposte (casa e trasferta) e il risultato finale

- **La tabella GOL:** Mappata in relazione uno-a-molti con il calendario tramite il MATCH_ID. Questa struttura permette di associare a ogni singolo incontro la granularità temporale e descrittiva delle reti, registrando il marcatore, il minuto esatto e la modalità realizzativa.

- **La tabella EVENTI:** Raccoglie la matrice multidimensionale delle statistiche di match, contenente dati relativi al possesso palla, i tiri totali, i tiri in porta, i calci d'angolo, i falli commessi, i fuorigioco e i cartellini gialli/rossi ricevuti, disaggregate per squadra.

- **La tabella RACCORDO:** Di fondamentale importanza, è stata introdotta per gestire la complessità delle relazioni topologiche, ovvero per analizzare i dati della stessa partita distinguendo la squadra di casa da quella in trasferta, e consentire il filtraggio incrociato dei dati senza generare ambiguità o percorsi di join circolari nel motore analitico di Power BI.

3.3 MODELLAZIONE STATISTICA E ANALISI TEMPORALE DELLE PARTITE

3.3.1 INDICATORI DI TENDENZA CENTRALE E MACRO-ANALISI DI LEGA

Prima di focalizzarsi sullo studio millimetrico del singolo avversario, la pipeline analitica esegue una macro-analisi descrittiva dell'intero campionato. Questa fase ha l'obiettivo di tracciare il profilo competitivo globale della lega, consentendo di comprendere se i dati registrati da una squadra rappresentino un'anomalia comportamentale o il riflesso di un trend collettivo.

Dal punto di vista narrativo e metodologico, questa sezione si propone di fornire una sintesi del torneo attraverso l'utilizzo di indicatori di tendenza centrale, come media e mediana, e di dispersione applicati alle matrici di prestazione.

L'analisi si sviluppa su tre direttrici principali:

1. **I volumi di produzione offensiva e l'efficienza di conversione:** Il sistema aggrega i dati relativi ai tiri totali, ai tiri in porta, ai tiri fuori e ai tiri respinti, posizionando ciascun club all'interno delle graduatorie ufficiali della lega *(Figura 3.8; Tabella 3.1)*. Tuttavia, il mero conteggio del volume di tiro rischia di essere fuorviante se disgiunto dalla qualità realizzativa. Per questo motivo, la pipeline mette in relazione il volume delle conclusioni con il numero effettivo di reti realizzate, calcolando l'indice di conversione, noto come *shooting accuracy*. Questo indicatore permette di distinguere tra squadre con alta realizzazione e squadre a "bassa efficienza" che, pur tirando frequentemente, difettano di precisione o di qualità nella scelta dello spazio di tiro.

```python
def calcola_efficienza_offensiva(filepath):
    # Caricamento dei fogli di lavoro relativi alle statistiche di match e ai gol realizzati
    df_stats = pd.read_excel(filepath, sheet_name='STATISTICHE COMPLETE x GIORNATA')
    df_gol = pd.read_excel(filepath, sheet_name='ELENCO GOL FATTI')

    # Aggregazione dei volumi di tiro totali e nello specchio per squadra
    volumi_tiri = df_stats.groupby('SQUADRA').agg({
        'TIRI': 'sum',
        'TIRI IN PORTA': 'sum'
    }).reset_index()
    volumi_tiri.columns = ['SQUADRA', 'TIRI_TOTALI', 'TIRI_IN_PORTA_TOTALI']

    # Conteggio dei gol effettivi realizzati da ciascuna squadra
    gol_realizzati = df_gol.groupby('SQUADRA').size().reset_index(name='GOL_FATTI')

    # Merge dei dataset e gestione di eventuali valori nulli
    df_analisi = pd.merge(volumi_tiri, gol_realizzati, on='SQUADRA', how='left').fillna(0)

    # Calcolo dell'indice di conversione (Shooting Accuracy): (Gol Fatti / Tiri Totali) * 100
    df_analisi['EFFICIENZA_CONVERSIONE_%'] = (df_analisi['GOL_FATTI'] / df_analisi['TIRI_TOTALI']) * 100

    # Ordinamento per efficienza di conversione decrescente
    ranking_efficienza = df_analisi.sort_values(by='EFFICIENZA_CONVERSIONE_%', ascending=False).reset_index(drop=True)
    ranking_efficienza['RANK'] = ranking_efficienza.index + 1

    print("--- RANKING EFFICIENZA OFFENSIVA E CONVERSIONE (TOP 10) ---")
    print(ranking_efficienza[['RANK', 'SQUADRA', 'TIRI_TOTALI', 'GOL_FATTI', 'EFFICIENZA_CONVERSIONE_%']].head(10).to_string(index=False))

    return ranking_efficienza
```

Figura 3.8

RANK	*SQUADRA*	*TIRI TOTALI*	*GOL FATTI*	*EFFICIENZA (%)*
1	*ASCOLI*	*344*	*63*	*18.32*
2	*CAMPOBASSO*	*307*	*49*	*15.96*
3	*RAVENNA*	*316*	*50*	*15.82*
4	*AREZZO*	*423*	*64*	*15.13*
5	*PINETO*	*287*	*42*	*14.63*

Tabella 3.1

Per ricavare i seguenti risultati è stata utilizzata la libreria Pandas, particolarmente utile per la manipolazione e l'analisi dei dati.

Dal foglio di lavoro "STATISTICHE COMPLETE x GIORNATA" del dataset proprietario è stata eseguita un'aggregazione dei tiri in porta e dei tiri totali in base alla relativa squadra.

Dal foglio di lavoro "ELENCO GOL FATTI" è stata eseguita la stessa aggregazione relativa ai gol fatti da ogni squadra.

Con i seguenti risultati è stato poi possibile calcolare l'efficienza (espressa in percentuale) come il rapporto tra gol fatti e tiri totali.

2. **L'analisi del possesso palla: Dominio territoriale vs verticalità effettiva:** Un altro snodo fondamentale della macro-analisi riguarda il tempo medio di possesso palla *(Figura 3.9; Tabella 3.2)*. La statistica descrittiva evidenzia la correlazione (spesso non lineare) tra il controllo prolungato del pallone e la produzione offensiva. Attraverso l'incrocio di questa metrica con i dati di passaggi e progressioni, la pipeline permette di smascherare il cosiddetto "possesso sterile", ovvero quelle situazioni in cui una squadra registra percentuali di possesso elevate nella propria metà campo o sulla trequarti, ma fallisce nell'occupazione degli spazi chiave e nella verticalizzazione, risultando di fatto inoffensiva per la retroguardia avversaria.

```python
def analisi_correlazione_possesso_tiri(filepath):
    # Caricamento del dataset delle statistiche complete per giornata
    df_stats = pd.read_excel(filepath, sheet_name='STATISTICHE COMPLETE x GIORNATA')

    # Aggregazione per squadra: calcolo delle medie stagionali per possesso e tiri
    sintesi_squadre = df_stats.groupby('SQUADRA').agg({
        'POSSESSO PALLA': 'mean',
        'TIRI': 'mean',
        'TIRI IN PORTA': 'mean'
    }).reset_index()

    sintesi_squadre.columns = ['SQUADRA', 'POSSESSO_MEDIO', 'TIRI_MEDIA_MATCH', 'TIRI_IN_PORTA_MEDIA']

    # Calcolo della correlazione di Pearson tra Possesso e Tiri Totali
    corr_tiri, p_value_tiri = stats.pearsonr(
        sintesi_squadre['POSSESSO_MEDIO'],
        sintesi_squadre['TIRI_MEDIA_MATCH']
    )

    # Calcolo della correlazione di Pearson tra Possesso e Tiri in Porta
    corr_porta, p_value_porta = stats.pearsonr(
        sintesi_squadre['POSSESSO_MEDIO'],
        sintesi_squadre['TIRI_IN_PORTA_MEDIA']
    )

    print("--- ANALISI STATISTICA: POSSESSO VS PRODUZIONE OFFENSIVA ---")
    print(f"Correlazione di Pearson (Possesso vs Tiri Totali): r = {corr_tiri:.3f} (p-value: {p_value_tiri:.4f})")
    print(f"Correlazione di Pearson (Possesso vs Tiri in Porta): r = {corr_porta:.3f} (p-value: {p_value_porta:.4f})")

    # Ordinamento per possesso palla medio decrescente
    ranking_possesso = sintesi_squadre.sort_values(by='POSSESSO_MEDIO', ascending=False).reset_index(drop=True)
    ranking_possesso['RANK'] = ranking_possesso.index + 1

    print("\nRanking squadre per possesso palla medio:")
    print(ranking_possesso[['RANK', 'SQUADRA', 'POSSESSO_MEDIO', 'TIRI_MEDIA_MATCH']].head(10).to_string(index=False))

    return sintesi_squadre
```

Figura 3.9

RANK	*SQUADRA*	*POSSESSO MEDIO*	*MEDIA TIRI*
1	*ASCOLI*	*59.11*	*9.56*
2	*AREZZO*	*53.81*	*11.75*
3	*TERNANA*	*51.86*	*9.81*
4	*RAVENNA*	*51.22*	*8.78*
5	*JUVENTUS NEXT GEN*	*51.03*	*7.36*

Tabella 3.2

Correlazione di Pearson (Possesso vs Tiri Totali): r = 0.592 (p-value: 0.0075)

Correlazione di Pearson (Possesso vs Tiri in Porta): r = 0.666 (p-value: 0.0018)

Per ricavare i seguenti risultati sono state utilizzate le librerie Pandas, come nel caso precedente, e la libreria Scipy.stats, utile per eseguire analisi statistiche e lavorare con distribuzioni di probabilità in modo semplice ed efficiente.

Dal foglio di lavoro “STATISTICHE COMPLETE x GIORNATA” del dataset proprietario è stata eseguita un’aggregazione, in base alla relativa squadra, per calcolare la media del possesso palla e la media dei tiri totali nell’arco della stagione.

I risultati ottenuti sono stati poi ordinati in senso decrescente in base al possesso di palla medio.

Inoltre, sono state calcolate due correlazioni di Pearson: La prima tra possesso palla e tiri totali, la seconda tra possesso palla e tiri in porta.

Dai risultati ottenuti si è potuta osservare una chiara e significativa correlazione positiva tra il possesso palla e la produzione offensiva.

3. **Il profilo disciplinare e l'intensità dell'aggressività agonistica:** La macro-analisi estende il proprio focus anche agli aspetti comportamentali e disciplinari del campionato, elaborando i dati relativi ai falli commessi e ai cartellini (gialli e rossi) estratti dal foglio di lavoro del dataset proprietario "ELENCO CARTELLINI" (Figura 3.10; Tabella 3.3). L'obiettivo è delineare l'indice di aggressività strutturale della squadra e degli avversari: comprendere se una formazione difende attraverso un pressing ultra-offensivo e sistematico, che solitamente si traduce in un alto numero di falli spesi per interrompere le transizioni avversarie, o se adotta un blocco basso, attendista e orientato al controllo posizionale dello spazio.

```python
def analizza_profilo_disciplinare(filepath):
    # Caricamento dei fogli relativi a cartellini e statistiche complete per i falli
    df_cartellini = pd.read_excel(filepath, sheet_name='ELENCO CARTELLINI')
    df_stats = pd.read_excel(filepath, sheet_name='STATISTICHE COMPLETE x GIORNATA')

    # Aggregazione dei cartellini gialli e rossi per squadra
    sintesi_cartellini = df_cartellini.groupby('SQUADRA').agg({
        'CARTELLINI GIALLI': 'sum',
        'CARTELLINI ROSSI': 'sum'
    }).reset_index()

    # Aggregazione del totale dei falli commessi per squadra
    sintesi_falli = df_stats.groupby('SQUADRA')['FALLI'].sum().reset_index()

    # Unione dei dataset disciplinari
    df_disciplina = pd.merge(sintesi_cartellini, sintesi_falli, on='SQUADRA')

    # Calcolo del totale dei provvedimenti disciplinari
    df_disciplina['TOTAL_CARTELLINI'] = df_disciplina['CARTELLINI GIALLI'] + df_disciplina['CARTELLINI ROSSI']

    # Ordinamento in base al numero di falli commessi (indice di aggressività e intensità di pressing)
    ranking_aggressività = df_disciplina.sort_values(by='FALLI', ascending=False).reset_index(drop=True)
    ranking_aggressività['RANK_FALLI'] = ranking_aggressività.index + 1

    print("--- PROFILO DISCIPLINARE E AGGRESSIVITÀ (RANKING PER FALLI COMMESSI) ---")
    print(ranking_aggressività[['RANK_FALLI', 'SQUADRA', 'FALLI', 'CARTELLINI GIALLI', 'CARTELLINI ROSSI', 'TOTAL_CARTELLINI']].head(10).to_string(index=False))

    return ranking_aggressività
```

Figura 3.10

RANK	*SQUADRA*	*FALLI*	*CARTELLINI GIALLI*	*CARTELLINI ROSSI*
1	*AREZZO*	*590*	*100*	*3*
2	*LIVORNO*	*563*	*91*	*4*
3	*FORLI'*	*561*	*80*	*1*
4	*VIS PESARO*	*561*	*97*	*4*
5	*PERUGIA*	*560*	*84*	*3*

Tabella 3.3

Anche nel terzo caso, per ricavare i risultati è stata utilizzata la libreria Pandas. Dal foglio di lavoro “STATISTICHE COMPLETE x GIORNATA” del dataset proprietario è stata eseguita un’aggregazione dei falli commessi durante l’intero campionato, in base alla squadra.

Dal foglio di lavoro “ELENCO CARTELLINI” è stata eseguita la stessa aggregazione relativa ai cartellini gialli e rossi ricevuti da ogni squadra.

I seguenti risultati sono stati poi riordinati, in ordine decrescente, in base al numero di falli di commessi.

Attraverso questa macro-analisi descrittiva, lo staff tecnico non ottiene soltanto una classifica numerica fredda, ma una mappa di posizionamento contestuale, utile a comprendere i punti di forza strutturali e le vulnerabilità sistemiche della squadra rispetto agli standard dell'intero campionato di Serie C.

3.3.2 L’ ANALISI DELLE SERIE STORICHE E DELLE FINESTRE TEMPORALI CRITICHE

Il rendimento di una squadra di calcio nel corso di una partita non è distribuito in modo uniforme, ma risente fortemente di variabili dinamiche quali la stanchezza atletica, i cali di tensione psicologica, la gestione dei ritmi agonistici e l'impatto strategico dei cambi dalla panchina.

Per catturare e decodificare le motivazioni che alterano il rendimento della squadra, la pipeline analitica scompone la durata regolamentare e di recupero dell'incontro in sei intervalli temporali standard da 15 minuti ciascuno:

- **1’-15’**: I primi quindici minuti di gioco rappresentano la fase in cui più di tutte si nota la mentalità e l’approccio alla gara delle due squadre. La mancata concentrazione in questo primo periodo può condizionare gravemente tutto il susseguo della gara.

- **16’-30’**: Il gioco entra nel vivo, una volta superata la prima fase di studio da parte delle due squadre in campo. Le squadre sono entrambe alla ricerca della stabilità in campo, tuttavia se il risultato non si sblocca inizia ad emergere l’ansia o la fretta di voler segnare a tutti i costi, fattore che può portare ad errori tecnici non forzati.

- **31’-45’+**: In questa fase la fatica accumulata inizia a farsi sentire e le posizioni in campo solitamente tendono ad allungarsi. In questo momento le squadre tendono a farsi investire ancor di più dall’ansia del gol, oppure tendono al comportamento contrario, ovvero a voler mantenere stabile la situazione maturata fino a quel punto, congelando il risultato. Specialmente quest’ultimo comportamento può portare a cali psicologici che rischiano di produrre l’effetto contrario rispetto a quanto desiderato.

- **46’-60’**: Le squadre scendono nuovamente in campo, e torna per la seconda volta il problema dell’approccio alla gara. Questo è uno dei momenti in cui l’attenzione dovrebbe essere massima, in quanto l’eventuale squadra in vantaggio dovrebbe cercare di non avere cali di attenzione, di fronte ad un avversario determinato a colmare il divario del primo tempo.

- **61’-75’**: In questa fase la stanchezza fisica inizia a incidere pesantemente sulla lucidità e sui movimenti dei reparti. In questo momento gli allenatori pescano forze fresche dalla panchina per cambiare l’inerzia tattica o correggere i problemi emersi nella prima ora di gioco. Diversi problemi caratterizzano questa fase. Se da una parte i nuovi giocatori devono essere in grado di adattarsi subito al livello agonistico della partita, dall’altra parte i giocatori stanchi, specialmente se ammoniti, devono rimanere maggiormente concentrati per evitare di sprecare forze inutili o concedere la maggioranza numerica in campo. Si tratta senza dubbio del periodo in cui entrano in campo più variabili, e dove ogni allenatore si appella a tutte le risorse a sua disposizione.

- **76’-90’+**: L’ultimo quarto d’ora di gioco, accompagnato da un corposo recupero che solitamente supera i cinque minuti è spesso caratterizzato da una stanchezza al limite, dove i reparti tendono a sgranarsi completamente. In questi ultimi minuti le distanze saltano, e la partita si gioca spesso sui nervi, sul cuore e sulla capacità di attingere alle ultime riserve energetiche.

Attraverso l'interrogazione granulare dei dataset storici dei gol realizzati e subiti, il sistema elabora distribuzioni di frequenza temporale che consentono di narrare la cronologia delle criticità di una squadra. Questi aspetti si vanno ad integrare con quelli facenti parte delle caratteristiche tipiche di una partita, descritte nell’elenco precedente.

Nello specifico, questa modellazione temporale permette di evidenziare **la vulnerabilità nell'approccio e in chiusura di frazione.** L'analisi individua con precisione la presenza di picchi anomali di reti subite o realizzate nei primissimi minuti di gioco o a ridosso dell'intervallo.

Un'alta concentrazione di gol subiti in avvio di gara svela solitamente problemi cronici di approccio mentale, scarsa concentrazione iniziale o un'incapacità strutturale di reggere l'impatto con la pressione aggressiva della squadra avversaria.

Come descritto in precedenza, un altro aspetto fondamentale riguarda **la tenuta atletica.** Gli ultimi minuti della partita rivelano spesso le capacità della tenuta fisica e mentale della rosa.

La pipeline inoltre evidenzia se una squadra possiede la capacità di sbloccare o chiudere i match nel finale, sfruttando il calo degli avversari o la propria profondità della panchina, oppure se manifesta una fragilità sistemica, palesando blackout difensivi nei minuti di massimo sforzo agonistico.

L’ultimo aspetto rilevato riguarda **l'impatto della fase centrale del secondo tempo**, che rappresenta il quarto d’ora che va dal 61esimo minuto al 75esimo minuto. Questo intervallo coincide solitamente con il momento dei cambi tattici e delle sostituzioni. L'analisi temporale permette di valutare l'efficacia delle contromosse dalla panchina e di identificare eventuali zone d'ombra in cui la squadra avversaria perde compattezza strutturale.

3.4 DALLA MACRO-ANALISI ALLO SCOUTING TATTICO: LA STRUTTURA OPERATIVA DEL REPORT TATTICO

L'obiettivo applicativo e terminale dell'intera pipeline analitica è la traduzione del patrimonio informativo estratto dal dataset di campionato e dalle fonti esterne in un documento di sintesi immediatamente fruibile: il **Report tattico dell'avversario**, dove per avversario si intende la prossima squadra affrontata nel campionato.

Il documento non si configura come una semplice raccolta di dati grezzi, ma come un percorso strutturato diviso in sezioni logiche sequenziali, progettate per accompagnare l'allenatore e lo staff tecnico nella definizione della strategia di gara.

La prima sezione riguarda **il rendimento recente e storico**, ovvero il confronto tra le ultime cinque partite e disputate e l'intero campionato svolto *(Figura 3.11; Figura 3.12)***.** In questa parte vi è il confronto tra il trend a breve termine e il rendimento globale della stagione. Le ultime cinque partite vengono analizzate distinguendo l'andamento casalingo da quello in trasferta. Viene inoltre analizzato il trend dei punti conseguiti, la stabilità dei risultati e lo scostamento statistico rispetto alle medie stagionali complessive.

Ultime 5 partite in casa: Ternana

20/03/26	(8) Ternana Calcio	1:1	Sambenedettese (17)
08/03/26	(8) Ternana Calcio	0:1	Gubbio (9)
01/03/26	(8) Ternana Calcio	2:4	Campobasso (4)
23/02/26	(8) Ternana Calcio	2:1	Forli (15)
12/02/26	(8) Ternana Calcio	2:0	Ravenna Calcio (3)

Ultime 5 partite fuori casa: Ternana

08/04/26	(6) Juventus II	2:1	Ternana Calcio (8)
04/04/26	(12) Guidonia	1:1	Ternana Calcio (8)
14/03/26	(10) Vis Pesaro	1:0	Ternana Calcio (8)
04/03/26	(1) AC Arezzo	1:2	Ternana Calcio (8)
16/02/26	(5) ASD Pineto Calcio	0:3	Ternana Calcio (8)

Figura 3.11

RENDIMENTO delle partite durante l'intera stagione

	In casa	Fuori casa	
1	AC Arezzo	1:1	1:2
2	Ascoli	0:2	1:0
3	Ravenna Calcio	2:0	1:0
4	Campobasso	2:4	0:1
5	ASD Pineto Calcio	2:1	0:3
6	Juventus II	1:0	2:1
7	Pianese	**26/04/26**	0:0
8	Ternana Calcio	-	-
9	Gubbio	0:1	3:3
10	Vis Pesaro	1:1	1:0
11	Livorno	2:1	1:0
12	Guidonia	1:2	1:1
13	AC Carpi	2:0	0:1
14	Perugia	**12/04/26**	1:2
15	Forli	2:1	1:1
16	Sassari Torres	0:2	1:1
17	Sambenedettese	1:1	0:0
18	Bra	2:0	**19/04/26**
19	Pontedera	2:1	2:2

Figura 3.12

Nella seconda sezione vengono analizzate **le statistiche globali e il posizionamento di lega** *(Figura 3.13)***.** Nel report, l'avversario viene inserito all'interno delle graduatorie generali del campionato, evidenziando il suo posizionamento per gol fatti e subiti, percentuali di possesso palla, volumi di tiri totali e nello specchio, falli effettuati e cartellini ricevuti. Viene successivamente effettuato il confronto con la squadra A.C. Perugia Calcio.

STATISTICHE GLOBALI

La TERNANA è la 6° squadra per gol fatti durante l'arco dell'intero campionato, con 43 GOL FATTI, mentre il PERUGIA è al 11° posto con 35 gol fatti.

La TERNANA rappresenta la 9° migliore difesa del campionato, con 34 GOL SUBITI, 6 in meno del PERUGIA (40).

Detiene un POSSESSO PALLA medio del 53,0%. Ciò porta la TERNANA ad essere la 6°squadra con maggiore POSSESSO PALLA del campionato.

I dati sui TIRI la evidenziano come una squadra molto forte nei TIRI TOTALI (2° posto con 412 tiri) a fronte dei 409 del PERUGIA.

È inoltre al 6° posto per tiri in porta (132) al 1° posto per tiri fuori (224)

Il PERUGIA è la 1°squadra con più FALLI dell'intero campionato (544), mentre la TERNANA è al 18° posto con 412 FALLI.

In tale ottica si sottolinea che la TERNANA ha ricevuto 56 cartellini (54 Gialli e 2 Rossi), a fronte dei 78 ricevuti dal PERUGIA. La TERNANA è la squadra con meno cartellini ricevuti di tutto il campionato.

La TERNANA è al 10° posto per numero di FUORIGIOCO TOTALI con un totale di 48 fuorigioco nell'intero campionato. (Dato da tenere in considerazione. Dato che inoltre risulta essere identico a quello del PERUGIA.)

Con 2 RIGORI SBAGLIATI, la TERNANA rappresenta la squadra con più errori dagli 11 metri.

Figura 3.13

Nella terza sezione vengono riportati i **punti di forza e le vulnerabilità strutturali dell'avversario** *(Figura 3.14; Figura 3.15; Figura 3.16)***.** Nella parte centrale del report vengono descritte le modalità realizzative ricorrenti dell'avversario, che possono avvenire con marcatori specifici o con un determinato fondamentale di gioco, come ad esempio il tiro da fuori area.

Subito dopo vengono svelate le debolezze difensive più evidenti, facendo sempre riferimento ad un determinato periodo della partita oppure ad uno specifico fondamentale di gioco.

Migliori Marcatori TERNANA (Gol totali 39).

- Dubickas E. → 9 gol
- Pettinari S. → 6 gol
- Ferrante A. → 5 gol
- Kerrigan L. → 3 gol

*** Va inoltre preso in considerazione che vi siano 4 giocatori fermi a quota 2 gol

Figura 3.14

ANALISI GOL FATTI TERNANA.

0-15: 4

16-30: 6

31- 45+: 6

46-60: 7

61-75: 9***(Dato da tenere fortemente in considerazione)

76-90+: 7

Tot: 39

ANALISI GOL SUBITI TERNANA.

0-15: 6

16-30: 3

31- 45+: 7 ***(Dato da tenere in considerazione)

46-60: 7 ***(Dato da tenere in considerazione)

61-75: 5

76-90+: 6

Tot: 34

Figura 3.15

ANALISI MODALITA' GOL FATTI TERNANA.

Autogol: 2

Azione Solitaria: 1

Calcio d'angolo: 2

Calcio di punizione: 1

Calcio di rigore: 2

Colpo di testa su cross: 5

Tiro da fuori area: 11 ***Dato da tenere fortemente in considerazione

Tiro in area: 9

Tiro su cross: 6

Tot: 39

ANALISI MODALITA' GOL SUBITI TERNANA

Azione Solitaria: 1

Calcio d'angolo: 2

Calcio di punizione:1

Calcio di rigore: 1

Colpo di testa su cross: 5

Tiro da fuori area: 5

Tiro in area: 9

Tiro su cross: 6

Tiro su filtrante/sponda:4 ***Elevato rispetto alle squadre analizzate finora.

Tot: 34

Figura 3.16

3.5 L’INTEGRAZIONE DEI LARGE LANGUAGE MODELS (LLM) E IL MODULO DI NATURAL LANGUAGE GENERATION (NLG)

3.5.1 IL PROBLEMA DELLA SINTESI E DELLA LATENZA COGNITIVA NEL CALCIO PROFESSIONISTICO

Uno dei colli di bottiglia operativi più critici all'interno di una società calcistica professionistica è rappresentato dalla forte asimmetria temporale tra la mole di dati grezzi producibili dai sistemi di analisi e il tempo effettivo a disposizione dello staff tecnico per assimilarli. In regime di preparazione settimanale di una gara ufficiale, l'allenatore e i suoi collaboratori operano sotto una cronica pressione temporale.

Se da un lato le tabelle numeriche, i fogli di calcolo strutturati, come Il dataset proprietario PREDICTO 2526.xlsx, e le dashboard visive di Power BI offrono un rigore analitico ineccepibile, dall'altro richiedono un'interpretazione cognitiva complessa e dispendiosa in termini di tempo.

Il rischio concreto è il cosiddetto **sovraccarico cognitivo**, che si genera quando vengono riportate troppe informazioni numeriche slegate tra loro che rallentano la traduzione del dato in una contromossa tattica efficace sul rettangolo verde.

3.5.2 L’ARCHITETTURA DI NLG TRAMITE GEMINI PRO

Per abbattere questa barriera e azzerare la latenza decisionale, la pipeline sviluppata in questo lavoro introduce un livello avanzato di *Natural Language Generation* (NLG) basato sull'utilizzo di Large Language Models (LLM), nello specifico integrando **Gemini Pro** tramite l'ecosistema di programmazione in Python e le API ufficiali di Google GenAI.

A differenza dei sistemi di chat interattivi tradizionali, questo modulo agisce come un **generatore automatico di report testuali strutturati**, progettato per sintetizzare l'enorme mole di dati numerici raccolti nelle sezioni precedenti del

progetto, quali statistiche globali, trend delle ultime partite, distribuzioni temporali e metriche posizionali relative ai gol fatti e subiti.

Il flusso operativo si sviluppa in tre fasi ingegneristiche sequenziali.

Nella prima fase vi è **l'aggregazione e l'ingestion dei KPI**, dove gli script Python interrogano i dataset strutturati, estraendo e aggregando gli indicatori chiave relativi alla squadra avversaria.

Nella seconda fase avviene l'**ingegnerizzazione del prompt**. È in questo momento che i dati numerici strutturati vengono incapsulati all'interno di un template di input fortemente guidato. Al modello viene assegnato un ruolo ben definito e rigorosi vincoli di stile e contenuto. Nel caso in questione, il prompt istruisce l'API di Gemini Pro ad agire come un *data analyst* professionista al servizio di una squadra di Serie C, imponendo una suddivisione logica della risposta in due blocchi predefiniti e prescrittivi:

- Il blocco di analisi descrittiva ("***Cosa evidenzia l'analisi***"): Deputato a narrare le criticità strutturali e i pattern comportamentali dell'avversario emersi dai dati.

- Il blocco di indirizzo strategico ("***Al Perugia si consiglia***"): Incaricato di tradurre le analisi statistiche in indicazioni pratiche e operative per lo staff tecnico in vista delle partite.

Nell'ultima fase avviene **la Generazione del report tattico PDF**. L'API elabora la richiesta in modo asincrono e restituisce una sintesi testuale discorsiva, pulita e priva di sovraccarichi numerici superflui. Questo output testuale viene acquisito dallo script di generazione dei report e iniettato direttamente all'interno della sezione conclusiva del documento PDF destinato all'allenatore.
Per comprendere al meglio il funzionamento del NLG, viene riportato di seguito un esempio esplicativo *(Figura 3.17, Figura 3.18, Figura 3.19)*.

COSA EVIDENZIA L'ANALISI

Squadra propositiva ma poco concreta sotto porta. La Ternana è:

- **2ª per tiri totali (412)**
- **6ª per tiri in porta (132)**
- **6ª per gol fatti (43)**

Questo evidenzia una squadra che produce moltissimo volume offensivo, ma con una conversione non sempre efficiente. Grande produzione, ma finalizzazione migliorabile.

Possesso palla elevato e in crescita, ma non sempre incisivo.

- **53% possesso medio (6ª nel campionato)**
- **59,2% nelle ultime 5 (dato altissimo)**

Il dato indica una squadra che controlla le partite, ma che fatica a trasformare il dominio territoriale in vantaggi concreti.

Crescita offensiva nelle ultime partite (nonostante i risultati negativi). Dati recenti:

- **tiri: 11 (vs 9,8 media)**
- **tiri in porta: 5,4 (vs 4,5)**
- **calci d'angolo: 5,4 (vs 4,8)**

Questo evidenzia una maggiore pressione offensiva ed una squadra più presente nella metà campo avversaria, ma non accompagnata da risultati (0 vittorie nelle ultime 5).

Forte pericolosità tra il 60' e il 75'. Distribuzione gol fatti:

- **9 gol tra 61-75 (dato più alto)**

Indica una squadra molto efficace nella fase centrale del secondo tempo, probabilmente legata a ritmo, cambi e calo avversario.

Figura 3.17

Grande utilizzo del tiro da fuori area. Modalità gol:

- **11 gol da fuori area (dato nettamente più alto)**
- **9 gol in area**

Questo evidenzia una squadra che calcia molto dalla distanza, con centrocampisti e trequartisti coinvolti, ma meno dominante dentro l'area.

Difesa discreta ma vulnerabile su cross e situazioni dinamiche, come evidenziato dai **34 gol subiti (9ª difesa)**
Tuttavia vi sono alcune criticità:

- **5 gol di testa su cross**
- **6 gol su cross**
- **4 su filtranti/sponde (dato alto)**

Ciò indica difficoltà nella lettura delle palle laterali e nelle situazioni dinamiche in area.

Momenti critici ben definiti. Distribuzione gol subiti:

- **7 gol tra 31-45+**
- **7 gol tra 46-60**

Questo evidenzia cali di concentrazione tra fine primo tempo e inizio secondo tempo, momenti chiave della partita.

Trend negativo recente e fragilità difensiva.

- **0 vittorie nelle ultime 5**
- **gol subiti in tutte le ultime 5**

Nonostante buoni numeri offensivi, emerge una fase negativa legata soprattutto alla tenuta difensiva e mentale.

Figura 3.18

Squadra estremamente disciplinata.

- **2°Squadra con meno falli del campionato ed una delle squadre con meno cartellini del campionato (56 totali)**

Questo indica un approccio molto pulito e controllato, ma potenzialmente poco aggressivo nei duelli.

Linea offensiva attiva ma più volte in fuorigioco recentemente.

- **media fuorigioco ultime 5: 2,6 (in forte aumento)**

Segnale di maggiore ricerca della profondità, ma anche di sincronismi non perfetti.

AL PERUGIA SI CONSIGLIA

- **Colpire tra fine primo tempo e inizio secondo tempo (30'-60'):** Momento di maggiore vulnerabilità difensiva
- **Sfruttare i cross:** La Ternana soffre molto le palle laterali e i duelli aerei
- **Attaccare la profondità:** Difesa vulnerabile su filtranti e movimenti alle spalle
- **Pressare nei momenti giusti:** Il loro possesso è alto ma non sempre pericoloso, pertanto risultante essenziale chiudere le linee e ripartire velocemente
- **Calciare anche da fuori area:** soluzione efficace vista la tipologia di gol subiti
- **Aumentare l'intensità difensiva:** la Ternana è poco aggressiva ed un'arma ulteriore si potrebbe sviluppare vincendo nei duelli fisici
- **Sfruttare il momento negativo:** squadra in calo mentale, importante approccio forte alla gara

PUNTO DEBOLE PIÙ EVIDENTE TERNANA

Alta produzione offensiva ma bassa concretezza + vulnerabilità su cross e tra 30'-60' + fase difensiva fragile nelle ultime partite.

Figura 3.19

L'introduzione di Gemini Pro all'interno della pipeline di Data Science rappresenta una perfetta simbiosi tra il rigore della matematica e la pragmatica del campo.

Infatti, l'utilizzo di questo NLG ha permesso di **ridurre il carico cognitivo**, consegnando all'allenatore un'analisi testuale fluida e mirata, che evidenzia immediatamente i punti deboli da attaccare e le insidie da evitare.

Riducendo il carico cognitivo si è naturalmente ottenuta una **velocizzazione del decision-making**, infatti il tempo necessario per interpretare matrici statistiche complesse si è ridotto al minimo, consentendo allo staff di concentrare maggiori risorse sulla pianificazione degli esercizi specifici da svolgere durante le sedute di allenamento settimanali, piuttosto che sull'interpretazione del lavoro di analisi.

Avendo costruito un processo interamente automatizzato, ad ogni aggiornamento del dataset di campionato, la pipeline sarò in grado di generare nuovamente report tattici aggiornati per qualsiasi avversario della categoria. Ciò rende il progetto **riproducibile e scalabile**, essendo perfettamente applicabile anche alla successiva stagione di campionato o ad altre categorie.

CAPITOLO 4

4.1 INTRODUZIONE E VISIONE INGEGNERISTICA DEL SISTEMA

Nel panorama di una società di calcio professionistico o dilettantistico, l'efficacia dei modelli di Sport Analytics applicati (descritti nel Capitolo 3) poggia inevitabilmente sulla solidità dell'infrastruttura informatica di base.

La gestione disorganizzata delle anagrafiche, la dispersione delle scadenze medico-sanitarie, la frammentazione della logistica di magazzino e l'assenza di un tracciamento centralizzato delle gare ufficiali rappresentano criticità strutturali che ostacolano l'efficienza operativa dei club.

Il presente capitolo descrive la progettazione, l'architettura software e lo sviluppo implementativo del sistema gestionale full-stack denominato **gestionale_asd_calcio**.

La piattaforma va oltre la concezione di un normale schedario digitale, configurandosi come un ecosistema software avanzato, scalabile e multi-utente che integra segreteria sportiva, pianificazione agonistica, gestione logistica dei materiali e flussi di analisi dati.

4.2 DIAGRAMMA CONCETTUALE DELLE CLASSI DI DOMINIO

Prima di analizzare nel dettaglio le singole interfacce operative e i flussi transazionali della piattaforma **gestionale_asd_calcio**, risulta fondamentale definire l'architettura logica e concettuale che regola la persistenza dei dati. Nel contesto di un'applicazione web multi-tenant orientata alla gestione sportiva e all'analisi dati, la modellazione delle informazioni richiede un rigore formale capace di rispecchiare fedelmente le dinamiche reali di un club.

La struttura di backend, sviluppata tramite l'ORM di Django, organizza le entità di dominio in un database relazionale normalizzato. Per comprendere la complessità e l'interconnessione di questo ecosistema, viene illustrato il diagramma delle classi di dominio e lo schema delle relazioni logiche, evidenziando le cardinalità (associazioni 1:1 e 1:N) che legano i diversi moduli della piattaforma *(Figura 4.1)*.

Per la creazione del seguente diagramma delle classi di dominio è stata utilizzata una piattaforma gratuita ed open source, dal nome ***Draw.io*** [21]**.**

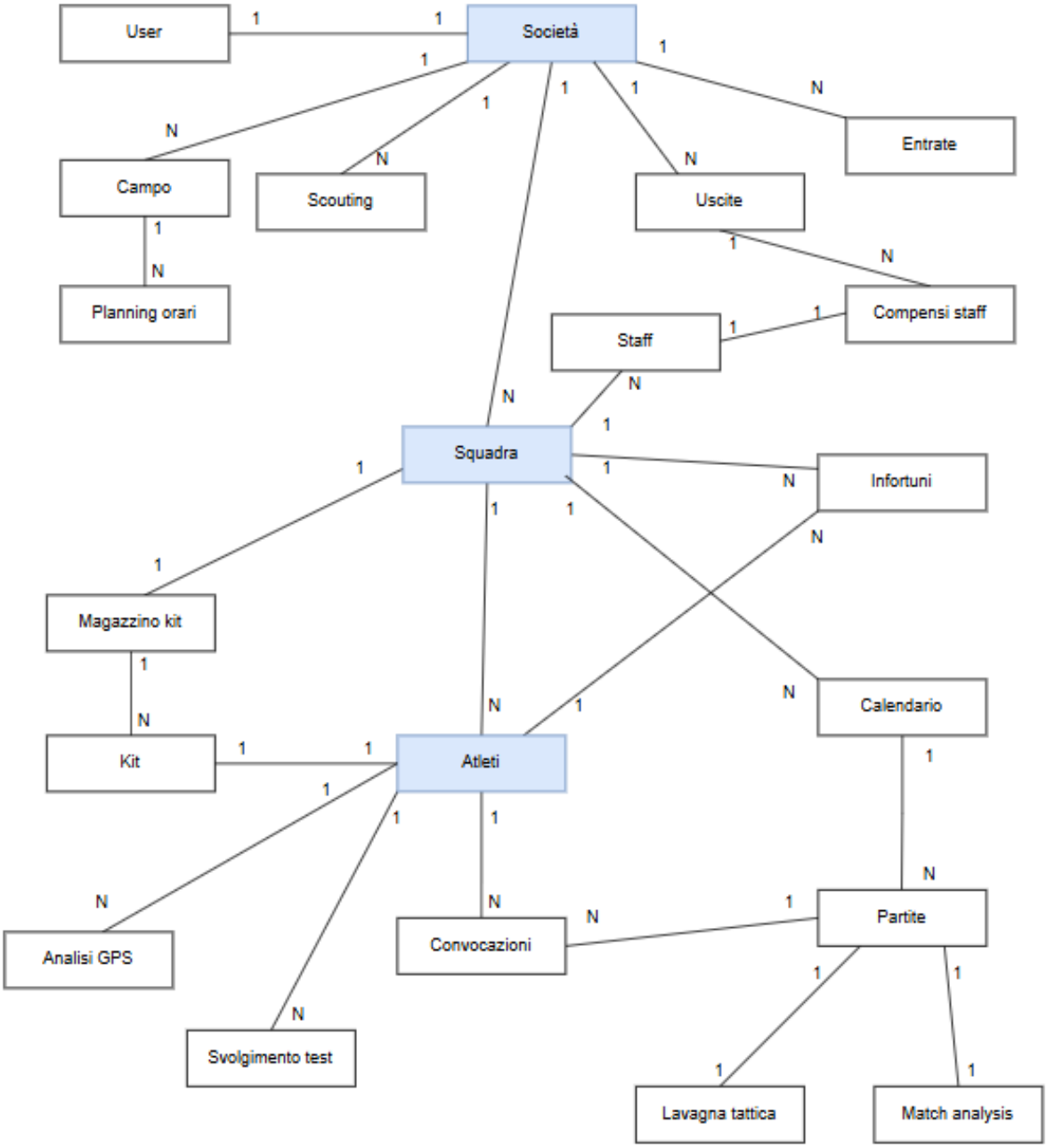


Figura 4.1

L'architettura si può suddividere in cinque macro-aree funzionali strettamente integrate.

La prima di queste è il **nodo centrale**, con l'entità *Società* che rappresenta la radice logica e il perimetro di isolamento del sistema. Ad essa si collega direttamente l'oggetto *User*, per la gestione delle credenziali di accesso. Dalla *Società* si diramano inoltre i rami operativi fondamentali, tra i quali figurano l'area

scouting, la gestione infrastrutturale degli impianti sportivi con i rispettivi orari e i flussi contabili di entrate ed uscite.

La seconda macro-area riguarda l'**area finanziaria e delle risorse umane**. In questa zona il ramo delle *Uscite* si collega biunivocamente ai *Compensi Staff*, che a loro volta attingono informazioni dall'entità *Staff*. Questa strutturazione garantisce la piena tracciabilità dei compensi e la conformità operativa con i requisiti normativi della Riforma dello Sport (D.Lgs. 36/2021).

La terza macro-area è quella dell'**area logistica e dei materiali**. Qui si specifica come ogni *Squadra* gestisce in modo centralizzato il proprio *Magazzino kit*, il quale si articola nei singoli elementi di abbigliamento (*Kit*), associati biunivocamente ai singoli *Atleti* tesserati, consentendo di monitorare in tempo reale le dotazioni assegnate e le eventuali carenze di scorta.

La quarta macro-area riguarda l'**area agonistica e il calendario**. La *Squadra* rappresenta l'insieme dei singoli *Atleti*. All' entità *Squadra* è collegato inoltre il *Calendario*, che governa la pianificazione delle singole partite.

L'ultima macro-area è rappresentata dall'**area di sport analytics**. Gli *Atleti* rappresentano l'entità cardine a cui afferiscono i moduli di sport analytics, come il modulo *Svolgimento test*, relativo ai test svolti dagli atleti durante la stagione; il modulo *Analisi GPS*, per l'analisi dei GPS degli atleti durante allenamenti e/o partite; il modulo *Infortuni* per il monitoraggio degli atleti che hanno riportato infortuni durante la stagione.

Sul fronte tattico, le *Partite* dialogano direttamente con i moduli *Lavagna tattica* e *Match analysis*, che forniscono sia uno strumento visivo pre-gara, sia uno strumento di analisi della partita stessa dal punto di vista del gioco.

4.3 MODELLAZIONE RELAZIONALE E SCHEMA DELLE CLASSI DI DOMINIO (3NF)

La persistenza strutturata delle informazioni all'interno del sistema **gestionale_asd_calcio** è affidata a un database relazionale, gestito tramite SQLite in ambiente di sviluppo e predisposto per MySQL in produzione, la cui progettazione logica rispetta rigorosamente i canoni della **terza forma normale (3NF)**.

Risulta opportuno precisare che la strutturazione di tale base dati non introduce ulteriori fonti di dati esterne rispetto ai flussi informativi e ai dataset analitici definiti nel Capitolo 3, limitandosi a definire l'architettura relazionale e i modelli di dominio necessari alla movimentazione operativa delle entità.

4.3.1 SCHEMA LOGICO ED ENTITÀ DI DOMINIO DEL SISTEMA

Per rispondere ai requisiti di integrità referenziale ed esplicitare la struttura relazionale dell'applicazione, viene di seguito presentato lo schema logico del database (*Figura 4.2*). Il diagramma evidenzia le tabelle, le chiavi primarie (PK), le chiavi esterne (FK) e i principali attributi descrittivi che compongono il dominio di **gestionale_asd_calcio**:

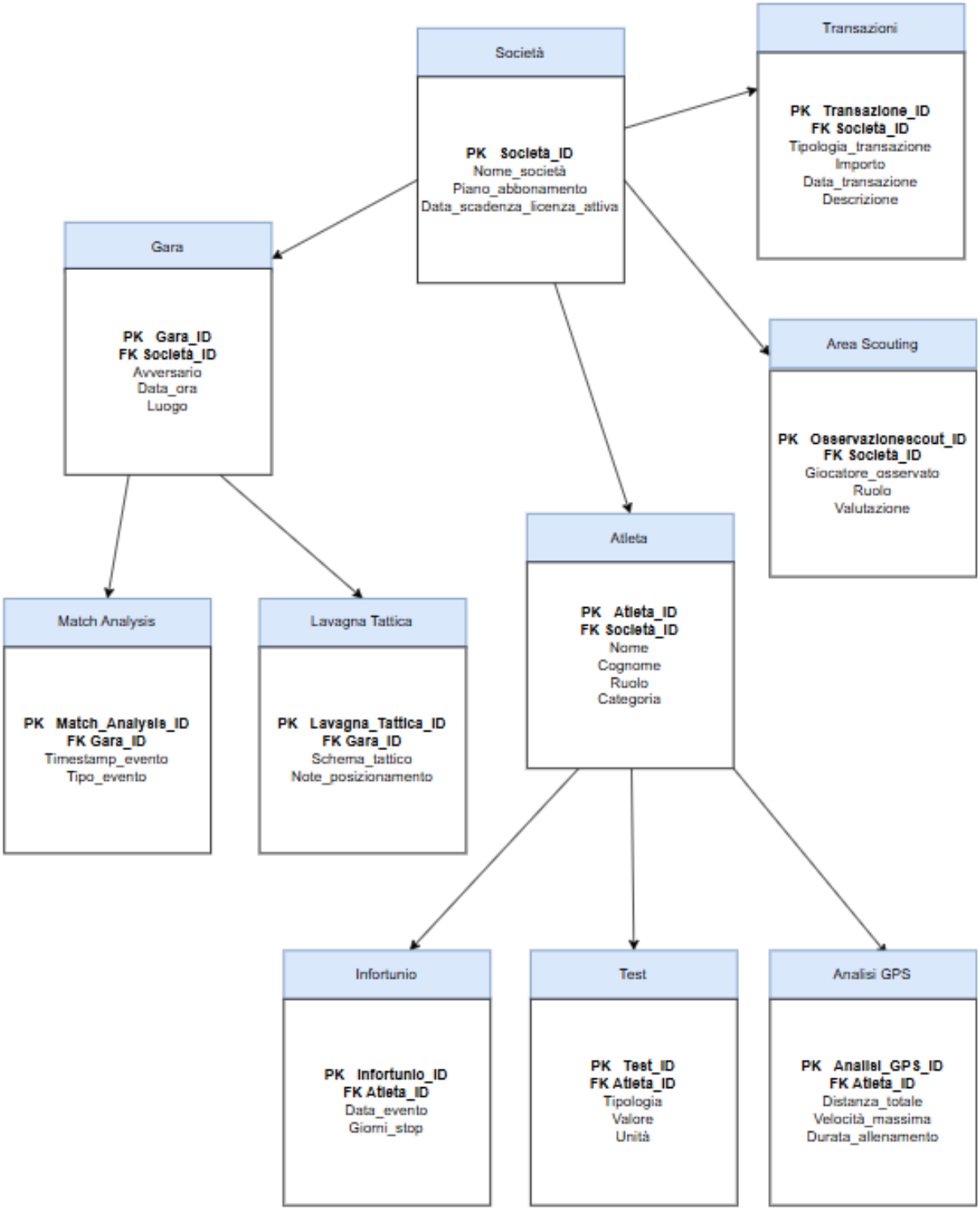

Società
PK Società_ID
Nome_società
Piano_abbonamento
Data_scadenza_licenza_attiva
Transazioni
PK Transazione_ID
FK Società_ID
Tipologia_transazione
Importo
Data_transazione
Descrizione
Gara
PK Gara_ID
FK Società_ID
Avversario
Data_ora
Luogo
Area Scouting
PK Osservazionescout_ID
FK Società_ID
Giocatore_osservato
Ruolo
Valutazione
Atleta
PK Atleta_ID
FK Società_ID
Nome
Cognome
Ruolo
Categoria
Match Analysis
PK Match_Analysis_ID
FK Gara_ID
Timestamp_evento
Tipo_evento
Lavagna Tattica
PK Lavagna_Tattica_ID
FK Gara_ID
Schema_tattico
Note_posizionamento
Infortunio
PK Infortunio_ID
FK Atleta_ID
Data_evento
Giorni_stop
Test
PK Test_ID
FK Atleta_ID
Tipologia
Valore
Unità
Analisi GPS
PK Analisi_GPS_ID
FK Atleta_ID
Distanza_totale
Velocità_massima
Durata_allenamento

Figura 4.2

4.3.2 PRINCIPI DI NORMALIZZAZIONE E VANTAGGI ARCHITETTURALI (3NF)

Nel contesto di un'applicazione web multi-tenant che gestisce dati eterogenei come quelli di anagrafiche sportive, scadenze sanitarie, calendari agonistici, logistica e flussi contabili, l'adozione di uno schema normalizzato in 3NF riveste un'importanza fondamentale per prevenire anomalie strutturali. Formalmente, una base di dati si dice in Terza Forma Normale se:

- **Si trova in seconda forma normale (2NF):** È in prima forma normale (ogni attributo contiene valori atomici) e tutti gli attributi non chiave dipendono interamente dall'intera chiave primaria, ovvero non vi sono dipendenze parziali.

- **Assenza di dipendenze transitive:** Nessun attributo non chiave dipende da un altro attributo non chiave. In altre parole, ogni attributo non chiave dipende unicamente e direttamente dalla chiave primaria della tabella.

L'applicazione rigorosa della 3NF al progetto **gestionale_asd_calcio** ha permesso di conseguire i seguenti vantaggi ingegneristici:

- **Eliminazione delle ridondanze:** Ciascun'informazione, come ad esempio i dati anagrafici di un atleta, i dati di un campo sportivo o le categorie di spesa della società, risiede in un'unica tabella di riferimento, azzerando il rischio di duplicazioni che porterebbero a disallineamenti informativi.

- **Prevenzione delle anomalie di aggiornamento:** La modifica di un attributo, come nel caso del cambio di denominazione di una struttura sportiva, richiede un unico intervento di scrittura sul database, riflettendosi istantaneamente su tutte le entità correlate tramite chiavi esterne.

- **Integrità referenziale garantita:** L'uso sistematico di vincoli di integrità e chiavi esterne impedisce la creazione di record orfani o incoerenti. Così facendo non sarà possibile commettere errori, come potrebbe essere invece un movimento di magazzino per un atleta inesistente.

4.3.3 IMPLEMENTAZIONE PRATICA DEI MODELLI DI DOMINIO

All'interno del modulo ***models.py*** dell'applicazione ***licenze***, le classi di dominio implementano le seguenti relazioni strutturali normalizzate:

1. **Gestione del tenant e degli utenti (società e user):** L'architettura multi-societaria poggia su una relazione di dipendenza in cui ciascuna entità operativa del gestionale possiede una chiave esterna obbligatoria (ForeignKey) verso il modello radice della società o dell'utenza autenticata, garantendo l'isolamento logico dei dati, noto come Multi-Tenancy.

2. **Modulo agonistico, campi e planning (squadra, gara, campo sportivo, planning orario, convocazione):** La pianificazione delle attività prevede entità relazionali dedicate: ***campo sportivo*** censisce le strutture fisiche, mentre ***planning orario*** gestisce l'assegnazione temporale degli impianti alle differenti squadre. Le gare ufficiali e amichevoli (Gara) si collegano agli atleti attraverso la tabella di raccordo esplicita ***convocazione***, la quale modula la relazione molti-a-molti (*ManyToManyField*) memorizzando attributi specifici quali il numero di maglia, lo status in distinta (*TIT o PAN*) e il minutaggio effettivo.

3. **Modulo contabile e di magazzino (uscite, magazzino kit):** La contabilità interna traccia le uscite e i compensi dello staff tramite il modello **uscite** e il modello **compensi staff** (dotato di campi monetari ad alta precisione).

Parallelamente, il magazzino gestisce le giacenze attraverso ***magazzino kit*** (con distinzione per taglia e codice SKU).

4. **Modulo di sport analytics (nota scouting, moduli di telemetria e infortuni):** Le valutazioni qualitative espresse dagli osservatori sono strutturate nel modello ***scouting***, che correla l'atleta ai voti multidimensionali (Fisica, Tecnica, Mentalità) e all'autore della recensione. I dati analitici (**GPS, test funzionali e registri clinici degli infortuni**) sfruttano modelli relazionali dedicati per l'archiviazione e l'interfacciamento con la pipeline di Data Science.

4.4 ARCHITETTURA DELL'INTERFACCIA UTENTE E ORGANIZZAZIONE DELLA DASHBOARD CENTRALE

L'interfaccia utente del sistema gestionale "**gestionale_asd_calcio**" è stata progettata seguendo i principi del *Responsive Web Design* e della massima efficienza ergonomica per operatori di segreteria, direttori sportivi, amministratori di sistema e staff tecnico.

il layout della **dashboard centrale** si articola in componenti strutturali interconnessi.

4.4.1 LA SIDEBAR DI NAVIGAZIONE MODULARE E CONTROLLO DEGLI ACCESSI

La barra laterale sinistra *(Figura 4.3)* costituisce il menù di navigazione principale del sistema. Essa organizza le funzionalità in due macro-aree logiche:

1. **Area operativa standard:** Comprende i moduli anagrafici e gestionali di base (panoramica, gestione squadre, gestione atleti, magazzino kit, calendario gare, gestione campi, planning orari, entrate & rate, spese & uscite, area scouting, registro presenze, staff & tecnici). Ciascuna voce interroga i rispettivi modelli relazionali tramite viste dedicate basate su class-based views di Django.

2. **Area di sport analytics:** Una sezione che ospita i moduli avanzati integrati con la pipeline di Data Science descritta nel capitolo 3 (data analysis (GPS), lavagna tattica, registro infortuni, test atletici, match analysis). L'accesso a tali moduli è regolato da livelli di accesso diversificati associati al profilo operativo della società, consentendo di abilitare o restringere la consultazione delle funzionalità analitiche avanzate in base alle specifiche configurazioni dei permessi di sistema.

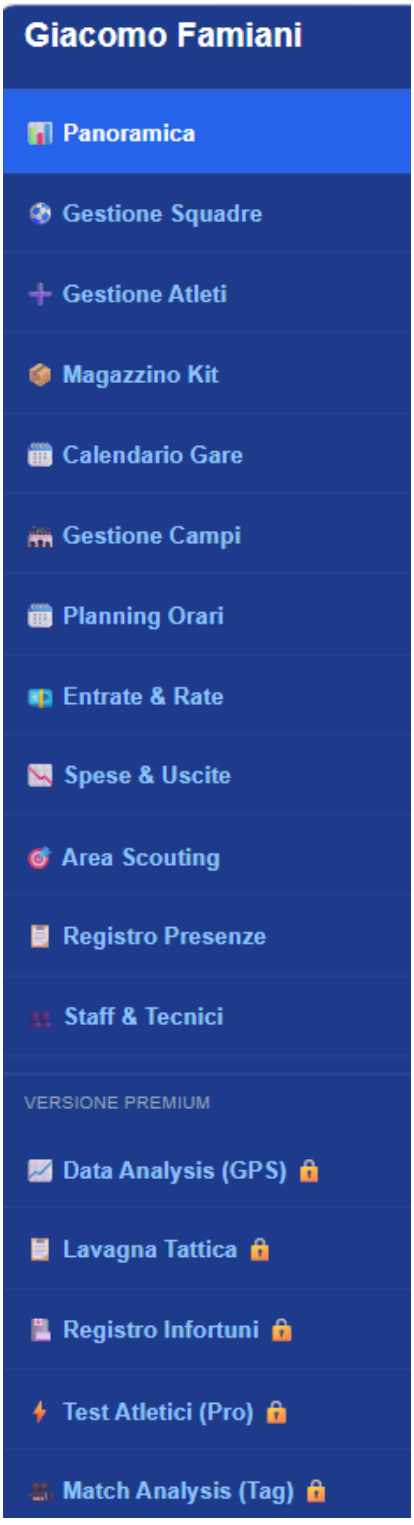

Giacomo Famiani
Panoramica
Gestione Squadre
Gestione Atleti
Magazzino Kit
Calendario Gare
Gestione Campi
Planning Orari
Entrate & Rate
Spese & Uscite
Area Scouting
Registro Presenze
Staff & Tecnici
VERSIONE PREMIUM
Data Analysis (GPS)
Lavagna Tattica
Registro Infortuni
Test Atletici (Pro)
Match Analysis (Tag)

Figura 4.3

4.4.2 LA BARRA DELLE AZIONI RAPIDE

Collocata in alto nell'area dei contenuti, la barra delle azioni rapide (Figura 4.4) offre collegamenti immediati per eseguire le operazioni più comuni di segreteria e di campo. Tra queste operazioni figurano:

- Crea squadra
- Aggiungi atleta
- Gestisci magazzino
- Pianifica gara
- Pianifica campo
- Registra entrata
- Registra spesa

Tali scorciatoie riducono drasticamente i passaggi di navigazione, consentendo all'operatore di inserire nuovi record senza abbandonare la schermata di riepilogo generale.

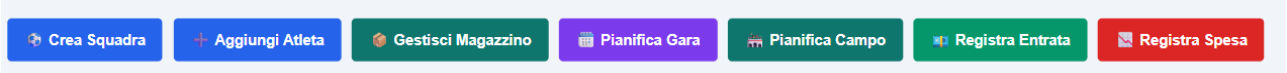


Figura 4.4

4.5 L'ARCHITETTURA SAAS MULTI-TENANT E IL PANNELLO DI CONTROLLO CENTRALIZZATO (DJANGO ADMIN)

Dal punto di vista dell'ingegneria del software, la scelta di sviluppare la piattaforma **gestionale_asd_calcio** adottando il paradigma **Software as a Service (SaaS)** non risponde a logiche commerciali, bensì a precisi requisiti architetturali e distributivi.

Un'architettura SaaS consente di centralizzare l'infrastruttura su una singola istanza applicativa, servendo molteplici organizzazioni sportive indipendenti attraverso un'unica codebase.

Questo approccio garantisce molteplici vantaggi strutturali rilevanti.

Uno di questi è sicuramente **l'isolamento logico dei dati**. La separazione delle informazioni tra diversi club (tenant) è garantita a livello di ORM tramite chiavi esterne obbligatorie che legano ogni entità operativa alla rispettiva società, preservando la sicurezza e la conformità normativa (GDPR).

Al contempo, **la manutenzione e gli aggiornamenti** della piattaforma vengono gestiti **in modo centralizzato**. Qualsiasi evoluzione della pipeline di Sports Analytics o patch di sicurezza viene rilasciata centralmente, rendendo le migliorie immediatamente disponibili per tutti gli utenti connessi senza la necessità di interventi di *deploy* locali.

Infine l'infrastruttura ottimizza il carico computazionale e previene la frammentazione dei dati tipica dei software tradizionali basati su architetture *on-premise*.

Per governare questo ecosistema centralizzato, il sistema sfrutta l'interfaccia di amministrazione nativa di Django (*Django administration*), opportunamente personalizzata ed estesa per supervisionare l'intera piattaforma (Figura 4.5).

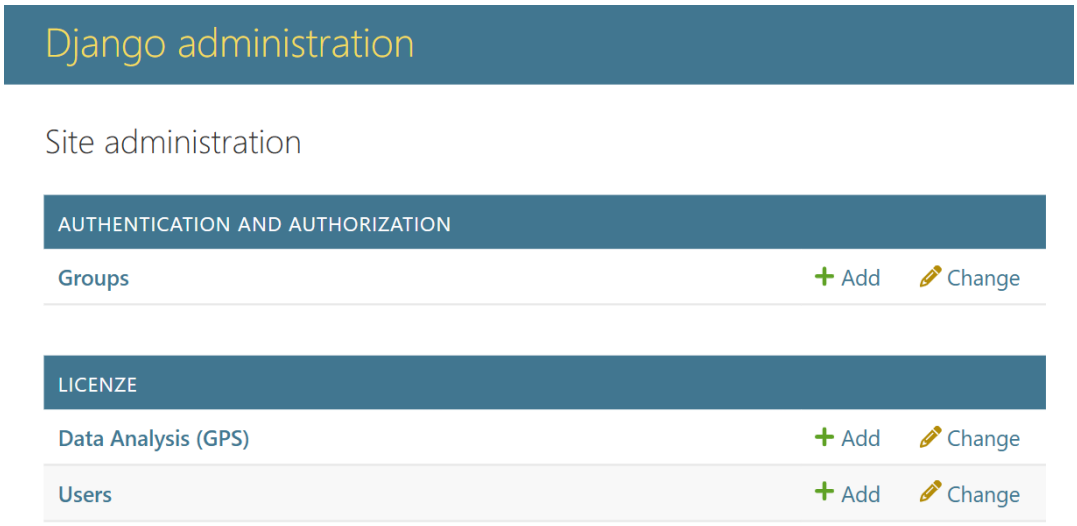


Figura 4.5

All'interno del pannello di controllo si distinguono due macro-sezioni.

- **GRUPPI**: Permette di aggregare gli utenti in categorie omogenee (es. *Dirigenti*, *Segreteria*, *Staff Tecnico*, *Magazzinieri*), assegnando a ciascun gruppo permessi collettivi di lettura, scrittura o eliminazione sui singoli modelli del database, evitando la configurazione manuale dei permessi per ogni singola utenza.

- **GOVERNANCE E GESTIONE DEGLI ACCESSI:** Questa sezione permette all'amministratore centrale di supervisionare i parametri strutturali dell'applicazione. Tra questi aspetti è fondamentale la sotto-sezione ***Users*** che gestisce l'anagrafica centralizzata di tutti gli utenti, consentendo di generare password cifrate, impostare PIN univoci e associare ciascun utente alla rispettiva società per preservare i confini del Multi-Tenancy.

È inoltre disponibile una piccola schermata con la cronologia delle ultime azioni svolte all'interno del programma.

4.6 CRUSCOTTO DEI KPI MULTIDIMENSIONALI E GESTIONE FINANZIARIA

Il cuore analitico della dashboard è rappresentato dal blocco dei **KPI Sintetici**, che aggrega in tempo reale le metriche finanziarie e anagrafiche della società sportiva:

1. **Bilancio Netto:** Indicatore finanziario calcolato in tempo reale sottraendo le uscite totali dalle entrate registrate.

2. **Entrate Totali:** Aggiornato tramite il modulo Entrate & Rate, traccia i flussi in entrata derivanti da quote associative, sponsorizzazioni o incassi botteghino.

3. **Spese Totali:** Gestito dal modulo Spese & Uscite, aggrega i costi vivi di gestione e logistica.

4. **Squadre / Categorie:** Numero di formazioni attive censite nella stagione corrente.

5. **Atleti Tesserati:** Consistenza numerica complessiva dei giocatori regolarmente tesserati e abilitati nel sistema.

4.7 ANALISI DETTAGLIATA DEI MODULI OPERATIVI E DELLE TABELLE DI GESTIONE

Avendo definito l'architettura di persistenza e le entità relazionali normalizzate in Terza Forma Normale, è possibile analizzare la traduzione applicativa di tali strutture all'interno della dashboard centrale. Questa interfaccia espone una serie di tabelle e form di riepilogo operativo che consentono il controllo diretto e la movimentazione delle entità di dominio modellate nel database. Ciascun modulo non costituisce un compartimento stagno, bensì si inserisce in una rete di relazioni funzionali e transazionali che garantiscono la coerenza globale del sistema

Oltre alle proprie funzionalità, vengono riportati anche i collegamenti con gli altri moduli, in modo da far comprendere in maniera esplicita, questa descrizione d’insieme che è stata descritta in precedenza.

4.7.1 MODULO “GESTIONE SQUADRE”

Il modulo **gestione squadre** è stato implementato mediante una vista basata su classi che interroga il modello Squadra applicando un filtraggio selettivo delle istanze basato sul tenant societario attivo. L'interfaccia grafica risultante espone in forma tabulare la categoria di riferimento, l'anno di competenza e il tecnico responsabile, sfruttando al contempo funzioni di aggregazione dinamica sul database per calcolare in tempo reale la consistenza numerica degli atleti tesserati *(Figura 4.6)*.

Collegamenti e funzionalità: Questo modulo rappresenta il nodo propedeutico e il primo passaggio logico obbligatorio dell'intera piattaforma: dalla definizione della squadra o categoria dipendono direttamente il tesseramento degli atleti, l'assegnazione degli slot orari nel *planning*, la programmazione degli impegni nel *calendario* e tutti i risultati che gli otterranno durante gli allenamenti, nei test e nelle partite della stagione.



Figura 4.6

4.7.2 MODULO “CALENDARIO”

Il modulo **calendario** gestisce la pianificazione temporale delle gare ufficiali, delle amichevoli e dei tornei svolti nell’arco della stagione. Ciascuna riga del calendario interroga la rispettiva entità relazionale riportando la tipologia di incontro, la squadra ospitante e la squadra ospite, la data, il luogo, ed un pulsante per la generazione asincrona delle distinte arbitrali in formato PDF *(Figura 4.7)*.

Collegamenti e funzionalità: Il calendario dialoga strettamente con il modulo di *gestione campi*, in modo da associare l’incontro all'impianto corretto evitando sovrapposizioni temporali. Il calendario funge inoltre da ponte diretto con il modulo di *match analysis*. La creazione di un evento di gara sblocca dinamicamente l'accesso al modulo di analisi della partita, con le sue rispettive funzionalità.



Figura 4.7

4.7.3 MODULO “GESTIONE CAMPI” & MODULO “PLANNING ORARI”

I seguenti moduli mappano le infrastrutture fisiche e gli impianti sportivi della società, quali campi in erba naturale o sintetica e palestre *(Figura 4.8)*.



Figura 4.8

Il sottosistema di planning incrocia i giorni della settimana con le fasce orarie e le squadre assegnate, interrogando le tabelle relazionali dedicate per prevenire conflitti di prenotazione sulle strutture *(Figura 4.9)*.

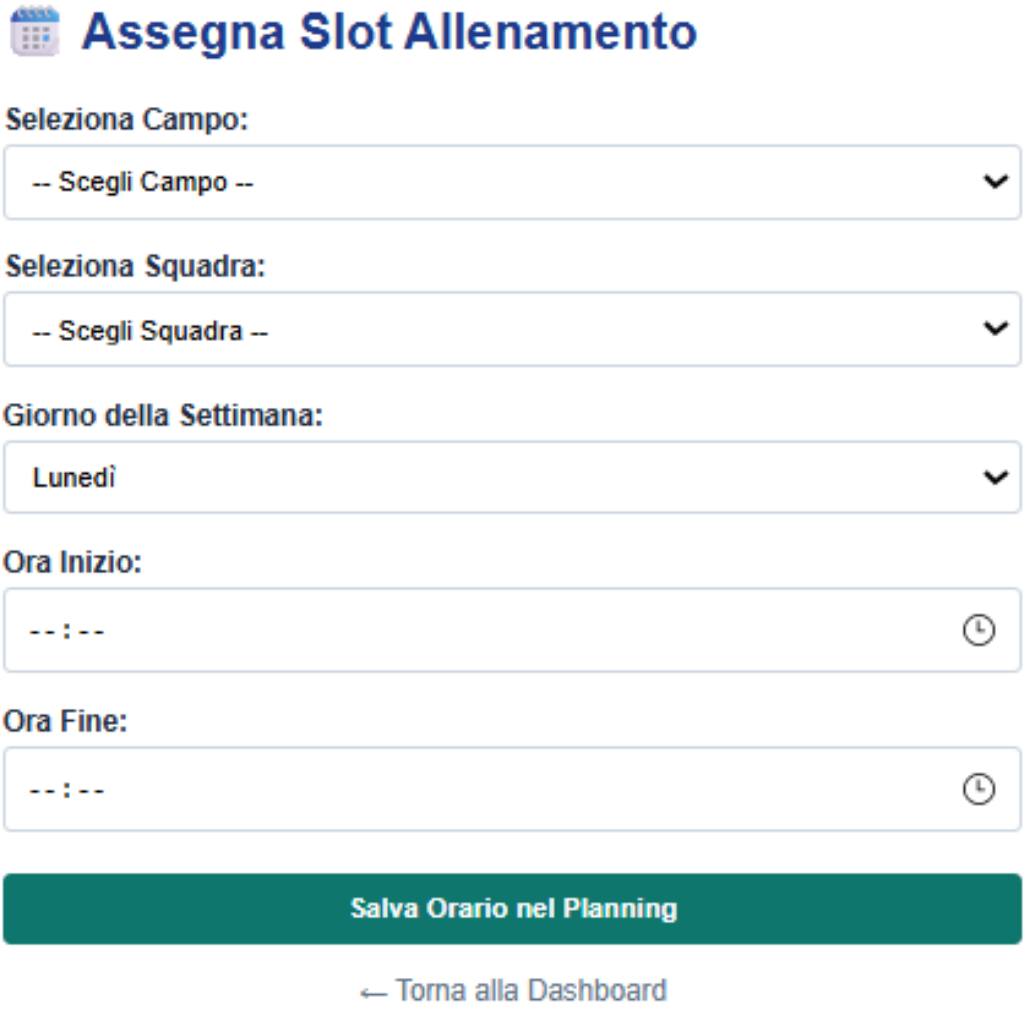


Figura 4.9

Collegamenti e funzionalità: Garantisce la sincronizzazione logistica tra la disponibilità degli impianti fisici e le necessità di allenamento delle singole formazioni definite nel modulo *Gestione Squadre*.

4.7.4 MODULO “MAGAZZINO KIT”

Controlla la disponibilità degli articoli sportivi suddivisi per codice SKU e taglia, interfacciandosi con un sistema di allertamento visivo basato sul confronto tra la giacenza reale registrata nel database e la soglia minima di scorta *(Figura 4.10)*.

Collegamenti e funzionalità: Questo modulo è strettamente integrato con il *modulo gestione atleti*, infatti durante il tesseramento o la revisione della scheda giocatore, il sistema verifica l'associazione diretta tra l'atleta e la taglia del kit assegnato, tracciando i flussi di consegna e verificando che il materiale sia stato consegnato al giocato.



Figura 4.10

4.7.5 MODULO “GESTIONE ATLETI”

Il modulo **gestione atleti** costituisce il registro centrale dei tesserati. Per ogni atleta vengono esposti i campi associati nel modello (squadra di appartenenza, ruolo, numero di maglia, matricola FIGC, stato dei kit) e il badge dinamico di controllo sanitario, calcolato in tempo reale confrontando la scadenza del certificato medico con l'orario di sistema. *(Figura 4.11)*.

Viene inoltre riportata una sezione relativa all’anagrafica del giocatore stesso.

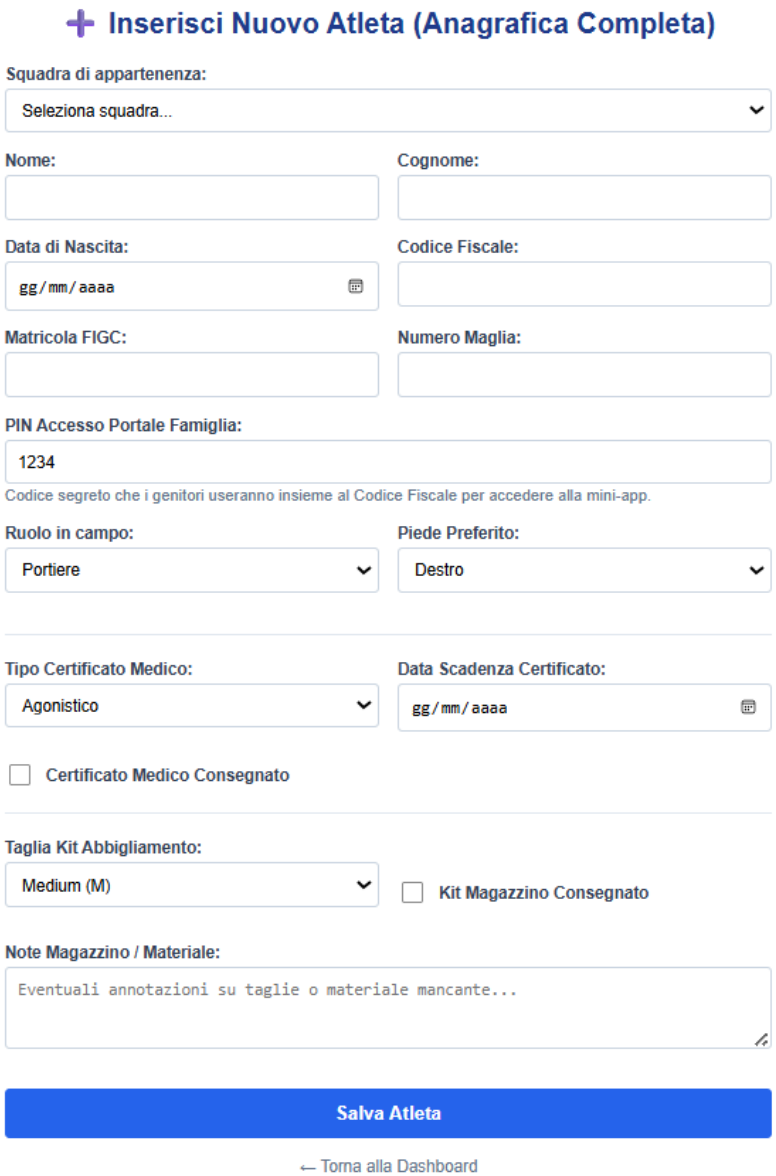


Figura 4.11

Collegamenti e funzionalità: È il modulo più interconnesso del sistema. alimenta le tabelle di *convocazioni* per le partite, riceve i dati di dotazione dal *magazzino kit*, attiva i blocchi di idoneità in base alle scadenze sanitarie (*certificato medico*), e costituisce la chiave d'accesso per il ***portale famiglia***, ovvero la mini-applicazione protetta da codice fiscale e PIN in cui atleti e genitori verificano i dati anagrafici, la situazione delle quote da pagare, le convocazioni alle partite, e dove riescono a controllare la presenza dei figli agli allenamenti *(Figura 4.12)*.

Sul fronte dello *sport analytics*, fornisce l'anagrafica di base per l'acquisizione dei dati provenienti dalle analisi dei GPS, dello storico clinico degli infortuni e dei test funzionali.

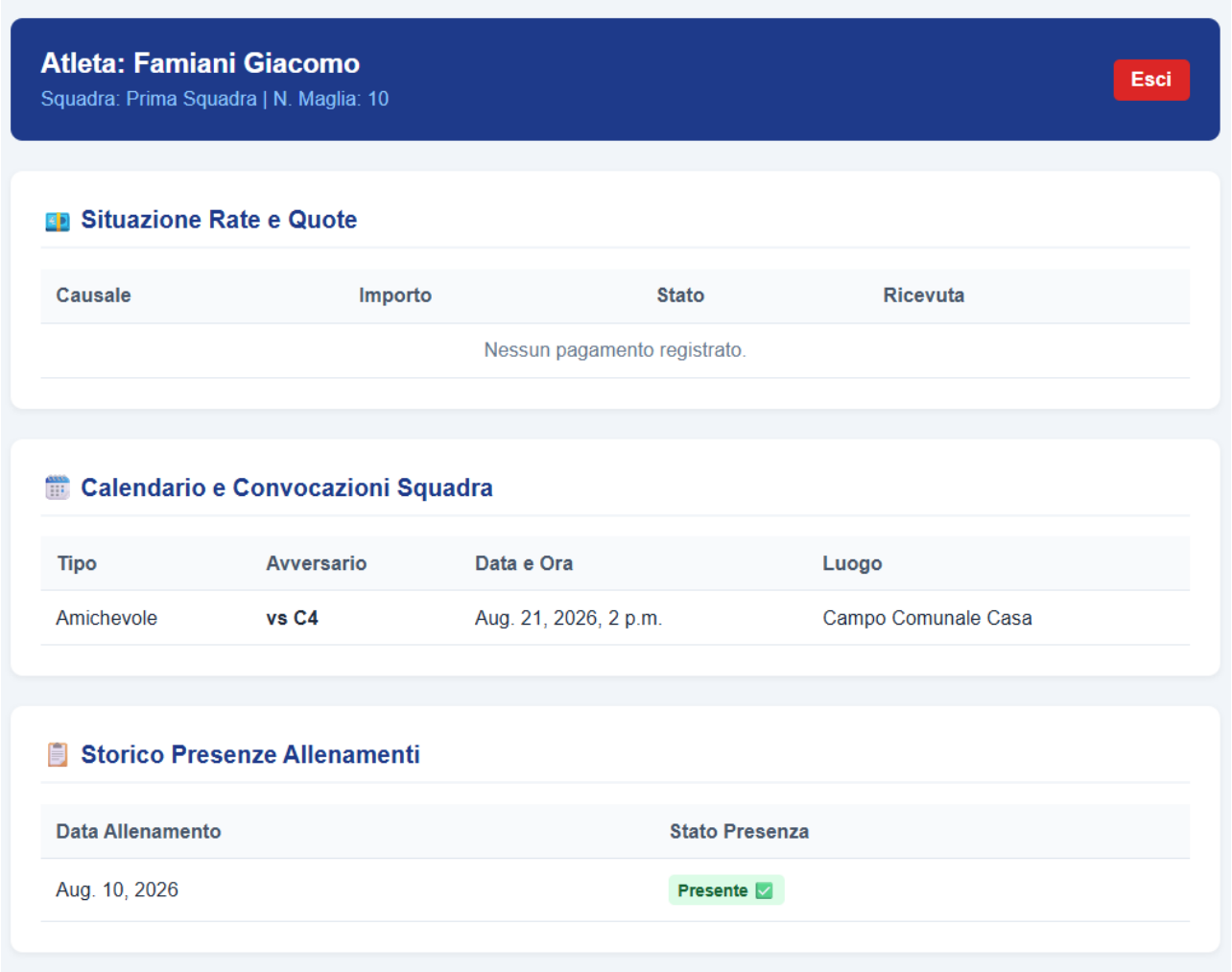


Figura 4.12

4.7.6 MODULO “AREA SCOUTING”

Il modulo **area scouting** rappresenta il ponte operativo tra le valutazioni qualitative espresse dallo staff tecnico e i dati quantitativi, registrando i parametri di Fisica, Tecnica e Mentalità *(Figura 4.13*).

Registrando il giocatore in questione, è possibile assegnare un voto da 1 a 10 per le tre categorie sopra-elencate, riportando anche il nominativo dello scoutman.

Collegamenti e funzionalità: Permette di tracciare l'evoluzione qualitativa dei calciatori monitorati. Il successivo step potrebbe essere quello di raggruppare i profili in cluster omogenei in base alle caratteristiche tecniche rilevate dagli osservatori.



Figura 4.13

4.7.7 MODULO “STAFF E TECNICI”

Il modulo **staff e tecnici** traccia i flussi finanziari e anagrafici relativi all'organigramma dei collaboratori, garantendo la corrispondenza tra i compensi erogati e gli accordi contrattuali memorizzati nelle tabelle di spesa. *(Figura 4.14*).

Collegamenti e funzionalità: Si interfaccia direttamente con il modulo contabile (*Spese & Uscite*) e riveste un ruolo chiave per assicurare la tracciabilità contabile e il pieno rispetto del **D.Lgs. 36/2021** (Riforma dello Sport) e dei successivi decreti integrativi in materia di compensi e tassazione del lavoro sportivo.



Figura 4.14

4.8 LA SEZIONE DI SPORTS ANALYTICS

All'interno dell'ecosistema software di **gestionale_asd_calcio**, l'area di ***sports analytics*** rappresenta il punto di contatto avanzato tra la gestione amministrativa quotidiana e le metodologie computazionali descritte nei capitoli precedenti.

Questa sezione è stata appositamente progettata e separata all'interno dell'architettura software per ospitare moduli ad alto contenuto analitico e predittivo, dedicati all'elaborazione dei dati di performance, alla match analysis e al monitoraggio clinico della squadra.

L'area specialistica si articola in cinque sotto-sezioni integrate, ciascuna caratterizzata da specifiche funzioni computazionali e profonde interconnessioni relazionali con il resto del database.

4.8.1 MODULO DATA ANALYSIS (GPS)

Il modulo **data analysis (GPS)** è stato sviluppato per la gestione e la visualizzazione dei carichi di lavoro telemetrici *(Figura 4.15*).

- **Architettura e ingestion dei dati**: Il modulo gestisce l'acquisizione e la visualizzazione dei carichi di lavoro telemetrici, ricevendo in input i file di log esportati dai sistemi GPS indossabili dai calciatori, come ad esempio accelerometri, giroscopi e ricevitori GNSS a 10Hz. Attraverso script di parsing automatizzati in Python, la piattaforma elabora i dati grezzi normalizzandoli per sessione, atleta e tipologia di seduta, che può variare in allenamenti di forza, scarico, tattica o partite ufficiali.

- **Metriche elaborate e relazioni**: Il sistema calcola indici avanzati di carico esterno e interno, tra cui la distanza totale percorsa, il volume ad alta intensità metabolica (HSR), il numero di accelerazioni e decelerazioni esplosive e il rapporto di carico acuto-cronico (ACWR). fondamentale per la prevenzione del rischio infortunistico. Dal punto di vista relazionale, i dati

si collegano direttamente all'entità atleta e stabiliscono un legame bidirezionale con il *registro infortuni*.

Tale legame risulta essere suggerito dalla letteratura, infatti molti infortuni che si verificano durante gli allenamenti, si possono ricondurre ad atleti che possedevano un indice ACWR superiore ad 1.5 e quindi superiore al valore soglia che definisce l'atleta in stato di pericolo infortuni [22,23].

- **Valore operativo**: Offre allo staff atletico cruscotti visivi longitudinali per monitorare lo stress fisico dei singoli reparti, consentendo di calibrare i carichi settimanali in funzione della prossimità con la gara successiva.

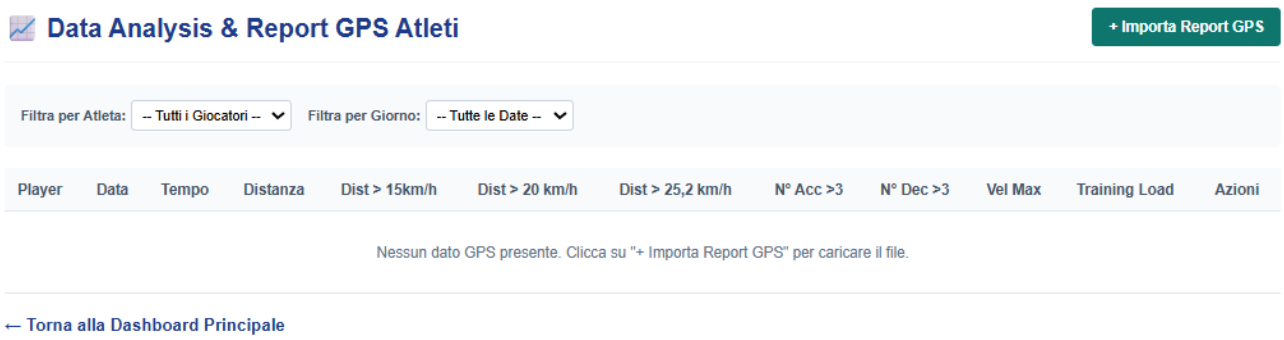


Figura 4.15

4.8.2 MODULO LAVAGNA TATTICA

Il modulo **lavagna tattica** è stato sviluppato mediante un ambiente grafico interattivo basato su HTML5, Canvas e librerie di rendering vettoriale. Il suo ruolo principale è consentire agli allenatori di digitalizzare la lavagna tradizionale, offrendo uno spazio di lavoro virtuale persistente. *(Figura 4.16*) [24].

- **Rendering grafico basato su HTML5 Canvas:** Il modulo è stato implementato sfruttando un elemento *canvas HTML5* gestito tramite un contesto di rendering bidimensionale in *JavaScript*. Questa scelta tecnica, contrapposta all'uso di elementi DOM tradizionali, consente il disegno diretto su pixel e garantisce una fluidità costante durante la movimentazione degli elementi e il tracciamento dei vettori.

- **Interazione e gestione degli oggetti di campo**: L'interfaccia risponde a eventi di input asincroni, come clic, trascinamento e comandi touch per il tablet. Le sagome dei calciatori, il pallone e i coni sono gestiti come oggetti dotati di coordinate spaziali (x, y) e metadati associati. Tramite un sistema di rilevamento delle coordinate, il client calcola il posizionamento e abilita il trascinamento fluido in tempo reale dell'entità selezionata.

- **Motore di disegno vettoriale:** Per tracciare linee di passaggio, corse e zone di pressing, il modulo si avvale degli strumenti di disegno messi a disposizione dal canvas. Il tratto libero impresso dall'allenatore viene convertito in linee e forme geometriche pulite, permettendo di alternare tratti continui, tratteggiati o frecce direzionali per rappresentare con precisione le consegne tattiche.

- **Persistenza dei dati e comunicazione con il Backend:** Lo stato corrente della lavagna, comprendente le posizioni delle sagome e le traiettorie tracciate, viene serializzato in formato *JSON*. Tramite chiamate asincrone, i

dati vengono trasmessi al backend di Django per essere salvati direttamente nel database relazionale, associandoli alla specifica seduta di allenamento o all'evento di gara.

- **Integrazione e flussi relazionali:** Questo strumento dialoga strettamente con il *Calendario*: la lavagna tattica si configura come il supporto operativo attraverso cui l'allenatore prepara e trasmette le indicazioni strategiche ai calciatori in vista della partita, durante le sessioni di analisi video e nei debriefing successivi agli incontri. La possibilità di salvare e archiviare digitalmente gli schemi costituisce il valore aggiunto che rende l'esercitazione persistente, consentendo di riprendere, modificare e riproporre le medesime situazioni tattiche all'interno della pianificazione agonistica settimanale.



Figura 4.16

4.8.3 MODULO REGISTRO INFORTUNI

Il modulo **registro infortuni** centralizza il monitoraggio sanitario della squadra permettendo allo staff medico di registrare e seguire l'intero percorso clinico di un atleta, dall'infortunio iniziale fino al completo recupero e al rientro in campo. *(Figura 4.18)*.

- **Come vengono salvati i dati nel database**: Il modulo si fonda su un modello di dominio dedicato, collegato direttamente alla scheda del singolo atleta tramite una relazione uno-a-molti. Il sistema salva informazioni in modo preciso e strutturato, riportando la data e l'ora dell'evento, la parte del corpo colpita, la causa dell'infortunio (distinguendo tra eventi traumatici o da sovraccarico), la prognosi stimata in giorni, lo stato di avanzamento del recupero e la data effettiva del via libera per il ritorno in gruppo. Inoltre, la validazione di questi dati gestisce direttamente le transizioni di stato: fino a quando l'atleta non viene dichiarato guarito, il sistema blocca automaticamente la sua convocabilità all'interno del modulo di *gestione gare*.

- **Come avviene l'incrocio tra l'infortunio e i dati GPS**: Come visto nel modulo precedente, i file grezzi dei dispositivi GPS non vengono salvati nel database per motivi di performance, ma le metriche elaborate, (come i volumi di corsa, le accelerazioni e l'indice di carico ACWR, sono memorizzate in modo normalizzato e collegate allo storico giornaliero del giocatore. Quando lo staff medico registra un nuovo infortunio muscolare, il programma sfrutta questa struttura relazionale per andare a ritroso nel tempo: automaticamente, il sistema esegue una ricerca incrociata tra le tabelle per leggere i dati GPS salvati nei 7 giorni precedenti per quello specifico atleta.
  A livello implementativo, la logica di questa interrogazione è gestita tramite un metodo dedicato nel backend che sfrutta l'ORM di Django. (*Figura 4.17*)

```python
from datetime import timedelta
from django.shortcuts import get_object_or_404
from .models import Infortunio, ModuloGPS

def analizza_carico_pre_infortunio(infortunio_id):
    # Recupera l'infortunio e l'atleta associato tramite chiave esterna
    infortunio = get_object_or_404(Infortunio, id=infortunio_id)
    atleta = infortunio.atleta
    data_infortunio = infortunio.data_evento

    # Calcola la finestra temporale dei 7 giorni precedenti l'infortunio
    data_limite = data_infortunio - timedelta(days=7)

    # Esegue la query filtrando per intervallo di date e per lo specifico atleta
    carichi_precedenti = ModuloGPS.objects.filter(
        atleta=atleta,
        data__gte=data_limite,
        data__lte=data_infortunio
    ).order_by('-data')

    return carichi_precedenti
```

Figura 4.17

Per quanto concerne il funzionamento del codice, la funzione riceve l'identificativo univoco dell'infortunio e utilizza la funzione di utilità *get_object_or_404* per recuperare l'istanza e il relativo atleta collegato tramite la chiave esterna. Sfruttando la classe *timedelta* del modulo standard *datetime* di Python, viene definita la finestra temporale di sette giorni antecedenti all'evento traumatico. Successivamente, l'ORM di Django traduce l'interrogazione in una query SQL strutturata sul modello *ModuloGPS*, filtrando i record in base all'appartenenza all'atleta e all'intervallo di date, ordinandoli infine in modo decrescente per data, per consentire allo staff medico di analizzare i carichi di lavoro precedenti.

- **Il valore operativo e la disponibilità della rosa**: Questo confronto automatico permette di capire se l'infortunio è arrivato dopo un picco improvviso di fatica o superando le soglie di sicurezza dell'indice di carico. In questo modo, lo staff medico e l'allenatore non vedono l'infortunio come un evento isolato, ma possono analizzarne le cause fisiche e calcolare in

tempo reale l'indice complessivo di disponibilità della rosa utile per valutare i rischi e pianificare le scelte di formazione.

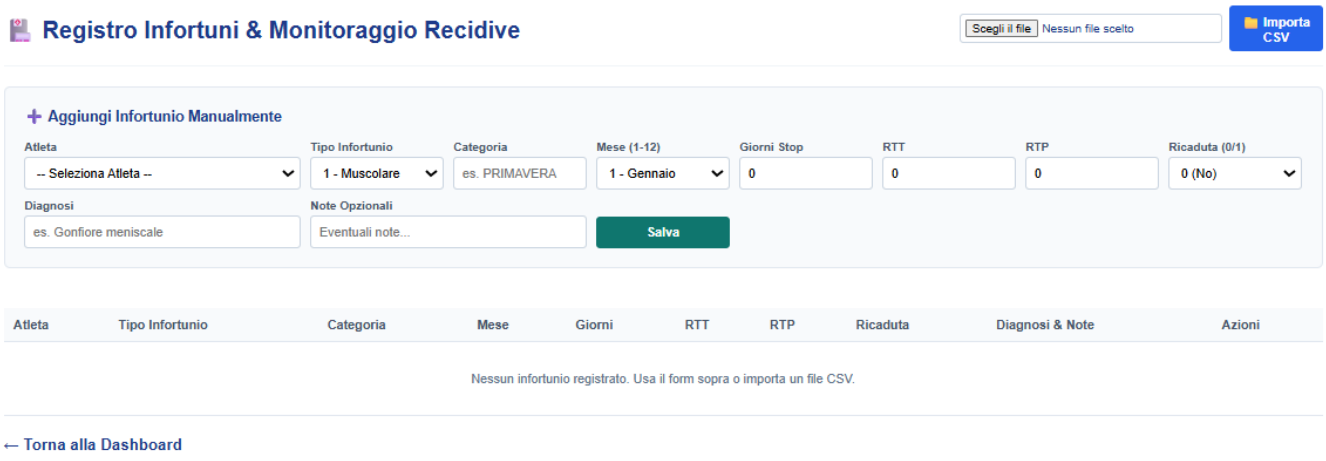


Figura 4.18

4.8.4 MODULO TEST ATLETICI

Il modulo dei **test atletici** è molto utile in quanto permette l'archiviazione e l'analisi longitudinale dei test di valutazione delle performance nei settori di interesse *(Figura 4.19*).



Figura 4.19

- **Valutazione funzionale, strutturazione dei dati e interfaccia:** Il modulo raccoglie i risultati dei test fisici periodici somministrati alla rosa, come ad esempio il test di soglia anaerobica, il test di salto verticale con pedana di Bosco oppure il test di agilità e forza massimale. L'architettura relazionale memorizza i test storicizzati per data e categoria, sfruttando una struttura dati flessibile che si adatta alle diverse unità di misura. Come evidenziato nelle interfacce, l'applicazione si articola in sezioni strutturalmente analoghe che differiscono unicamente per l'unità di misura dei valori di test (secondi oppure metri) e per la tipologia di prova eseguita in funzione della categoria scelta (Figura 4.20, Figura 4.21 e Figura 4.22).

- **Algoritmo di calcolo automatico dei trend e indicatori di Tendenza:** L'analisi della performance è automatizzata direttamente dalla logica di business del backend. All'inserimento di un nuovo test, un algoritmo interroga il database per individuare la rilevazione precedente dello stesso

atleta per la medesima prova, calcolando in tempo reale la variazione percentuale della prestazione.

$$\Delta\% = \frac{Valore\ attuale - Valore\ precedente}{Valore\ precedente} \times 100$$

Questo calcolo alimenta inoltre il campo descrittivo *Tendenza*, evidenziando immediatamente i tassi di miglioramento o le flessioni della condizione atletica.

- **Confronto analitico (curva standard vs media):** Oltre al trend individuale, l'interfaccia integra metriche di analisi collettiva. Il sistema mette a confronto il valore del singolo atleta sia con la **media di Squadra,** calcolata in tempo reale sui compagni di squadra, sia con una **curva standard di riferimento**. Quest'ultima non è una media mobile, bensì un profilo esogeno predefinito. I valori target ideali, o i benchmark storici stagionali, vengono memorizzati in apposite tabelle di configurazione filtrate per categoria, consentendo al backend di estrarre il parametro ideale atteso e di affiancarlo alla performance reale dell'atleta.

- **Valore operativo:** Questa combinazione di indicatori supporta il preparatore atletico nell'individuare tempestivamente i calciatori in ritardo di condizione (particolarmente indietro nei test rispetto al resto del gruppo) o soggetti a stanchezza cronica (giocatori con test che nel corso del tempo sono calati vertiginosamente), guidando con basi oggettive la personalizzazione e la calibrazione dei programmi di allenamento settimanali.

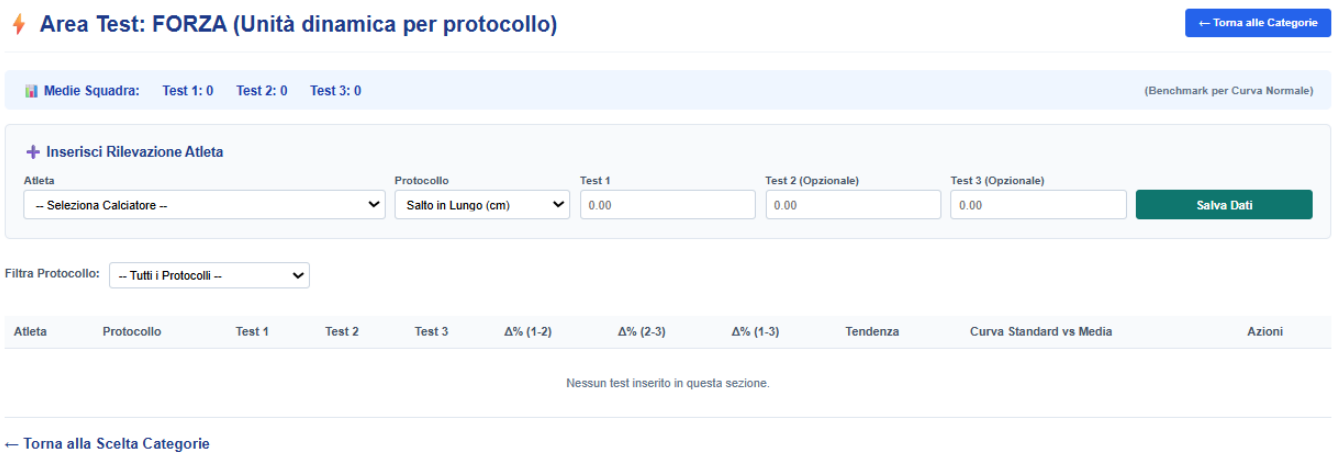


Figura 4.20



Figura 4.21

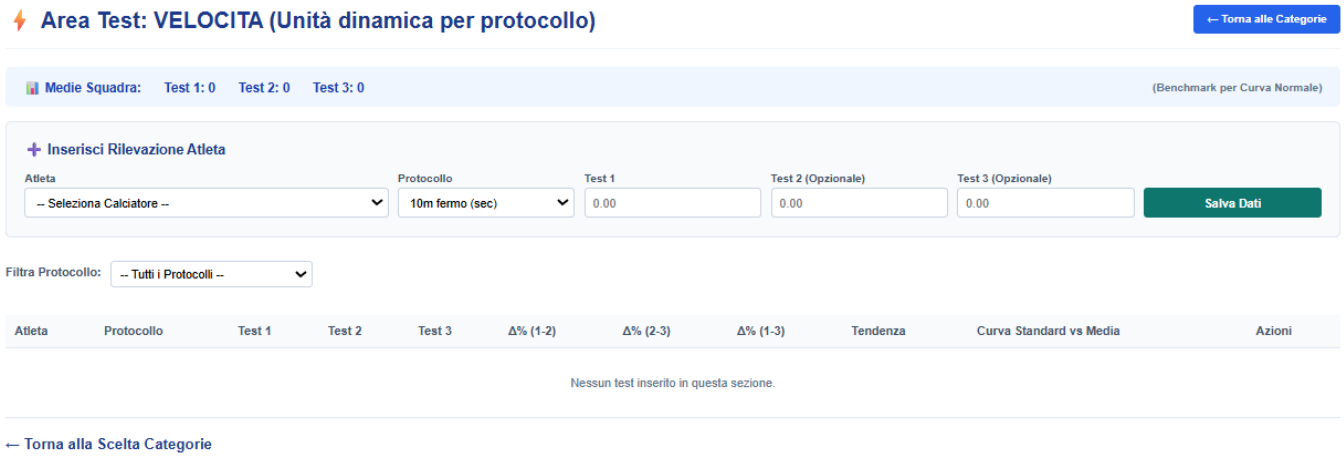


Figura 4.22

4.8.5 MODULO MATCH ANALYSIS

Il modulo **match analysis** riveste un'importanza strategica fondamentale nell'economia della piattaforma, poiché costituisce il ponte digitale tra l'archiviazione degli eventi di gara e il riscontro operativo sul campo.



Figura 4.23

- **Innesco operativo e controllo degli accessi:** A seguito della creazione e pianificazione di un evento nel modulo *Calendario (Figura 4.23)*, la dashboard principale visualizza dinamicamente un pulsante d'azione dedicato (etichettato come **ANALISI**) in corrispondenza della riga della partita Tale componente si attiva sulla base dei ruoli assegnati all'utente, sbloccando l'accesso all'interfaccia di codifica e sincronizzazione. *(Figura 4.24, Figura 4.25 e Figura 4.26)*.

- **Architettura del sistema di tagging e gestione degli eventi:** L'interfaccia di codifica temporale è stata sviluppata utilizzando componenti interattivi basati su JavaScript asincrono e gestori di eventi per l'interazione utente. L'analista interagisce con una plancia a opzioni preconfigurate, come ad esempio transizioni negative, palle inattive, tiri in porta o errori in costruzione. Durante la revisione del match, l'operatore immette manualmente all'interno dell'interfaccia i minuti e i secondi esatti del marcatore temporale, abbinandoli alla tipologia di evento corrispondente per la successiva memorizzazione strutturata all'interno del database.

- **Valore operativo e prospettive evolutive:** Pur non implementando algoritmi complessi di elaborazione automatica dei flussi, il modulo si fonda su un'architettura solida basata sul tagging manuale strutturato e sulla correlazione relazionale con i dataset esterni. Questo approccio consente allo staff tecnico di estrapolare rapidamente i segmenti delle situazioni tattiche analizzate nei report settimanali, velocizzando il feedback visivo da trasmettere ai calciatori durante le riunioni post-gara e pre-gara.

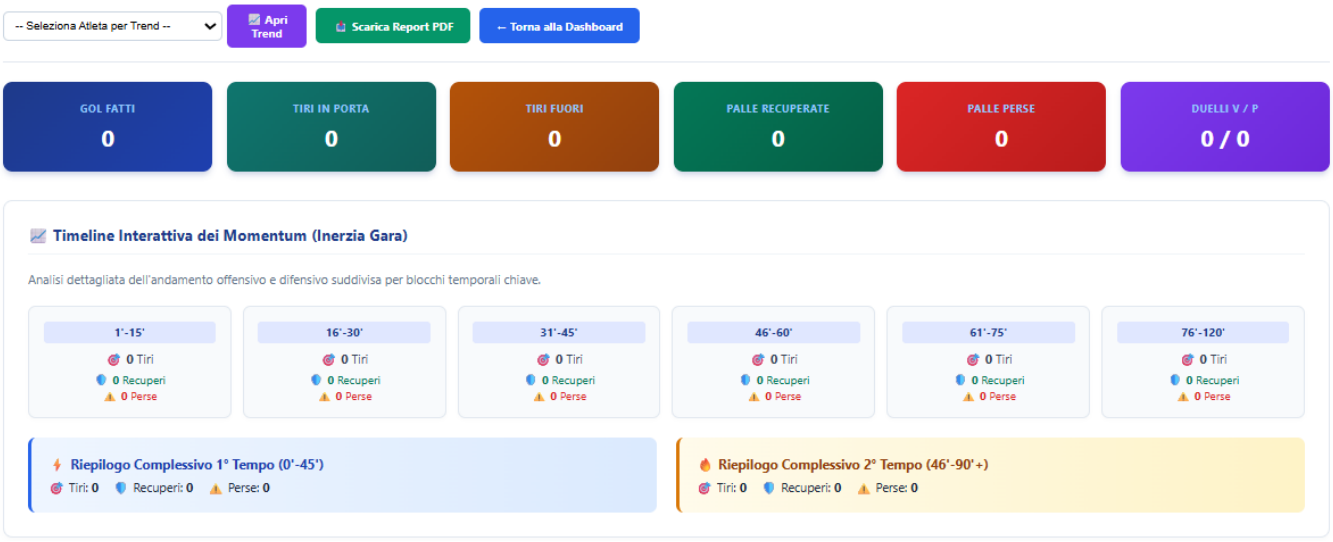


Figura 4.24

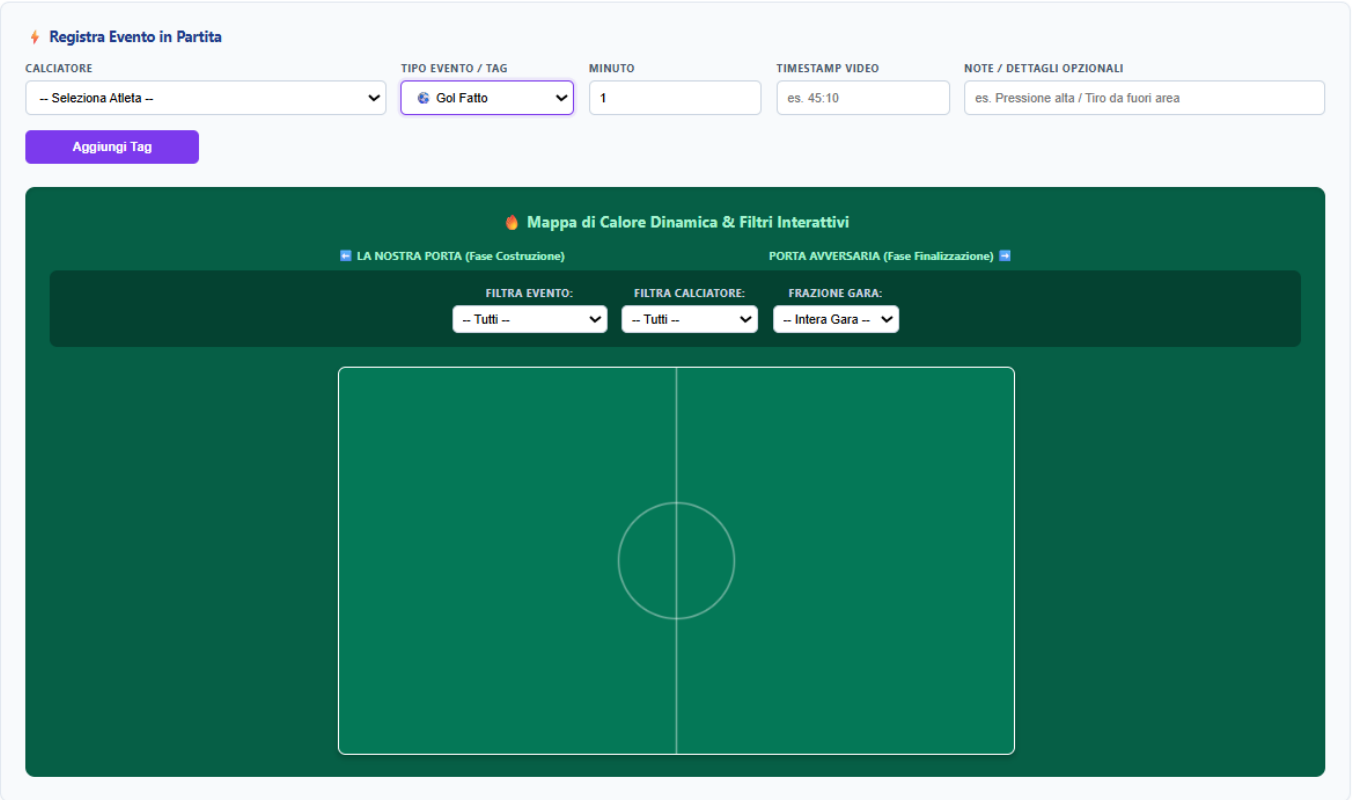


Figura 4.25

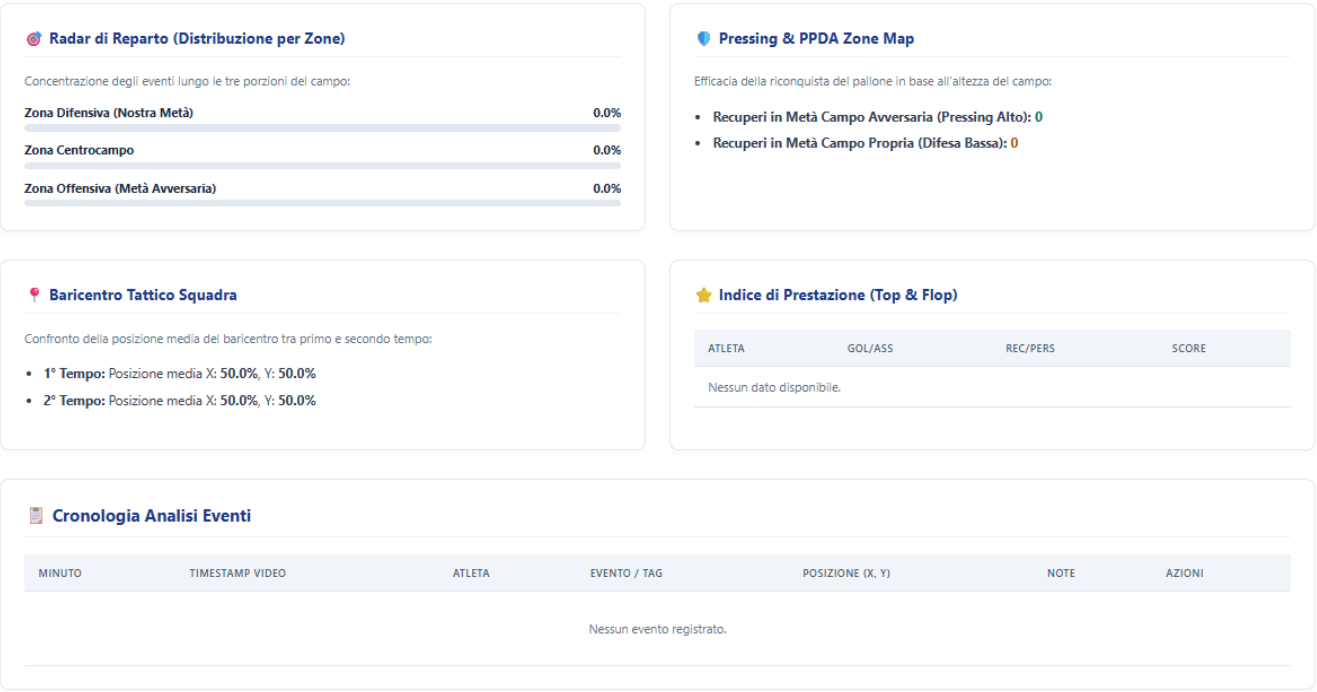


Figura 4.26

4.9 ARCHITETTURA SOFTWARE E ORGANIZZAZIONE DEI COMPONENTI

4.9.1 SCOMPOSIZIONE MODULARE E ALBERATURA DEL PROGETTO

L'architettura software di **gestionale_asd_calcio** si basa sul framework **Django (Python 3.x)** ed è organizzata all'interno del package principale di progetto (core_project), che gestisce le impostazioni globali (settings.py), le rotte principali (urls.py) e i punti di contatto con il server web (wsgi.py, asgi.py).

Il vero motore logico della piattaforma è racchiuso nell'applicazione dedicata chiamata **licenze**. Questa app segue rigorosamente il pattern architetturale Model-View-Template (MVT), suddividendo le responsabilità in file ben definiti:

- **models.py**: Contiene la definizione di tutte le classi di dominio relazionali (anagrafiche, gare, magazzino, contabilità, scouting e tracciamento infortuni).

- **views.py**: Gestisce le logiche di controllo, elaborando le richieste che arrivano dagli utenti e restituendo i dati dinamici da mostrare a schermo.

- **middleware.py**: Modulo customizzato responsabile dell'intercettazione delle richieste HTTP per la verifica in tempo reale dello stato della licenza e dei permessi per navigare.

- **urls.py**: Mappa i vari indirizzi web (URI) indirizzandoli dritti alle viste corrispondenti

- **Tree dei template (templates/licenze/):** I file HTML sono organizzati in modo pulito per separare le schermate standard da quelle avanzate.

Troviamo la dashboard principale (dashboard.html, portale_dashboard.html), i moduli per inserire o modificare i dati (aggiungi_atleta.html, aggiungi_gara.html, registra_pagamento.html, registra_spesa.html) e le pagine relative alla sezione di sports analytics (infortuni.html, lavagna_tattica.html, data_analysis.html, match_analysis.html, sezione_test.html), oltre alla schermata di blocco blocco_licenza.html per le utenze prive di abbonamento attivo.

4.9.2 CICLO DI VITA DELLA RICHIESTA http E CUSTOM MIDDLEWARE DI LICENZA

Quando un utente interagisce con la piattaforma, ogni singola richiesta HTTP attraversa una serie di passaggi in sequenza gestiti dai cosiddetti middleware:

1. **Protezione perimetrale e sicurezza (SecurityMiddleware, CsrfViewMiddleware):**

    - **Perché sono fondamentali**: Rappresentano la prima linea di difesa del sito. Il SecurityMiddleware aggiunge intestazioni di sicurezza fondamentali alla risposta HTTP, come la protezione contro l'esecuzione di script malevoli o l'incorporazione non autorizzata in iframe, mentre il CsrfViewMiddleware blocca i tentativi di falsificazione di richiesta inter-sito controllando che i moduli inviati provengano effettivamente da sessioni legittime tramite token crittografici.

    - **Perché servono proprio qui**: In un gestionale di questo tipo, questa sicurezza non è un dettaglio trascurabile. La piattaforma custodisce dati molto delicati: cartelle cliniche e infortuni, movimenti finanziari e contabili e segreti tattici di cui gli allenatori sono molto gelosi. Senza queste barriere, un malintenzionato potrebbe sfruttare una falla per compiere azioni al posto di un utente autorizzato, come ad

esempio, alterare i dati di bilancio o modificare lo stato clinico di un giocatore.

2. **Autenticazione e gestione sessioni (SessionMiddleware, AuthenticationMiddleware):**

- **Cosa si intende**: Quando si naviga sul sito, il browser invia un cookie cifrato che fa da "badge di riconoscimento". Il *SessionMiddleware* prende questo badge, va a leggere i dati salvati sul server e li deserializza, trasformandoli in formato leggibile dal programma ed inserendoli in *request.session*.
  Subito dopo, l'*AuthenticationMiddleware* usa quelle informazioni per capire esattamente quale utente del database sta effettuando l'accesso e lo assegna all'attributo *request.user*.

- **Utilizzo**: Si tratta del meccanismo che permette al sito di riconoscere subito l'utente, capendo al volo se si tratta di un allenatore, un medico o un preparatore atletico, così da mostrare solo le sezioni e i comandi permessi a tali figure (Si tratta di una funzionalità in fase di implementazione e prevede che, oltre alle società, anche lo staff disponga di diverse credenziali legate alla rispettiva figura e collegate ad alcune funzionalità specifiche della società stessa, in base a specifici permessi).

3.Il middleware personalizzato di licenza (licenze.middleware):

- Il middleware custom si mette in mezzo tra la richiesta dell'utente e le pagine avanzate dei moduli di sports analytics. Controlla lo stato della licenza e se l'abbonamento è scaduto o non dispone di tali privilegi, devia la navigazione verso la pagina di blocco. Se invece tutto è in regola, lascia passare la richiesta dritta verso la vista di destinazione, senza rallentare la navigazione.

4.10 LOGICA DI BUSINESS, FORM VALIDATION E TRANSAZIONI ATOMICHE (ACID)

La robustezza della logica applicativa implementata all'interno del modulo ***views.py*** e dei form di sistema si fonda su tre pilastri ingegneristici complementari:

- **Validazione semantica tramite model form:** Prima che qualsiasi dato immesso nei form di inserimento (es. *aggiungi_atleta.html, aggiungi_gara.html, registra_spesa.html*) venga persistito nel database, l'input dell'utente viene validato a livello applicativo attraverso classi ***model form*** personalizzate. Queste form applicano controlli di coerenza logica e temporale stringenti (ad esempio, verificando che le date dei certificati medici o le scadenze finanziarie rispettino i vincoli formali previsti dalla normativa e dalla logica di dominio).

- **Consistenza transazionale e proprietà ACID:** Per evitare *race conditions* e disallineamenti informativi in scenari di concorrenza multi-utente (ad esempio, registrazioni contabili simultanee o aggiornamenti di magazzino), il backend implementa blocchi transazionali espliciti mediante l'utilizzo del decoratore *@transaction.atomic*. Questo garantisce che un'intera sequenza di operazioni di scrittura sul database vada a buon fine solo se non si verificano eccezioni, preservando l'integrità del sistema.

- **Controllo della concorrenza tramite blocco pessimistico (select_for_update()):** Nelle operazioni più delicate, come la gestione delle giacenze o dei registri finanziari, l'ORM di Django mette un vero e proprio "lucchetto" temporaneo alla riga del database tramite la query SELECT ... FOR UPDATE. In questo modo, se un utente sta modificando un dato finanziario, nessun altro può leggerne o alterarne lo stato finché la transazione non è completata e salvata (*commit*).

CAPITOLO 5

L’implementazione di un’infrastruttura di Sports Analytics all’interno di un club professionistico di Serie C non si esaurisce nella correttezza computazionale degli script di elaborazione dati o nella pulizia dei dataset relazionali. Il vero banco di prova risiede nella validazione sul campo, ovvero nella capacità della piattaforma software di integrarsi fluidamente nei processi decisionali dello staff tecnico, riducendo la latenza tra l'acquisizione del dato grezzo e la stesura della strategia tattica settimanale.

In ambito accademico e professionale, la letteratura di riferimento evidenzia come il valore di un sistema di supporto decisionale, con particolare attenzione al caso in esame, non dipenda unicamente dalla complessità dei modelli statistici sottostanti, ma anche dalla sua capacità di inserirsi senza frizioni nei cicli cognitivi dell'allenatore.

Nel contesto del calcio professionistico moderno, caratterizzato da una mole crescente di informazioni eterogenee, il rischio concreto è quello di trovarsi di fronte a un’eccessiva quantità di dati grezzi o disorganizzati che, anziché chiarire lo scenario competitivo, andrebbero a generare un sovraccarico cognitivo, rallentando l'azione prescrittiva.

Pertanto il processo di validazione risulta fondamentale. La comprensione dei dati da parte dello staff tecnico ed i risultati ottenuti nel campo da gioco rappresentano una sinergia ideale, che è poi la base di quanto ipotizzato quando viene intrapreso un progetto di questo tipo.

5.1 L'USO DELLE PIATTAFORME E L'ABBATTIMENTO DELLA LATENZA DECISIONALE NEL CICLO SETTIMANALE

La gestione della settimana tipo all'interno di un club professionistico di Serie C è caratterizzata da una strutturale compressione temporale. Dalla conclusione dell'incontro domenicale alla rifinitura tecnico-tattica che precede la gara successiva, lo staff tecnico dispone di un intervallo estremamente circoscritto, che solitamente varia fra due o tre allenamenti, per metabolizzare l'esito della prestazione precedente, analizzare le caratteristiche dell'avversario di turno, individuare le vulnerabilità strutturali e trasmettere i concetti chiave al gruppo squadra, per cercare di trarne un vantaggio.

In questo scenario operativo, l'infrastruttura PREDICTO è stata concepita e ingegnerizzata per rispondere a un'esigenza fondamentale: ***L'abbattimento della latenza decisionale***.

La latenza decisionale rappresenta il tempo che intercorre tra l'avvenuta produzione del dato grezzo (la partita giocata dall'avversario) e la sua effettiva traduzione in una scelta strategica di campo (la contromisura tattica adottata dal Perugia).

La pipeline sviluppata riduce tale latenza attraverso un flusso operativo sequenziale ad alta efficienza:

- **Automazione dell'ingestion e del data wrangling (Python):** L'aggiornamento automatico dei dataset tabulari (PREDICTO 2526.xlsx) e l'esecuzione degli script di pulizia eliminano completamente la fase di inserimento manuale delle statistiche di match. I dati grezzi provenienti dai provider vengono normalizzati e convertiti in matrici relazionali pronte all'uso nel giro di pochi secondi.

- **Consultazione interattiva ed esplorativa (Power BI):** Il modello relazionale a stella consente all'allenatore e ai tecnici dello staff di interrogare visivamente i KPI storici e di campionato (possesso palla, conversion rate, flussi disciplinari) attraverso dashboard interattive. Ciò

permette di filtrare rapidamente le performance in casa o in trasferta senza la necessità di manipolare codice informatico o tabelle complesse.

- **Sintesi prescrittiva (Modulo NLG via Gemini Pro API):** Si tratta della funzionalità più apprezzata dall'allenatore e dai suoi collaboratori. In un contesto dinamico e bisognoso di informazioni rapide, ma precise, Il modulo di Natural Language Generation condensa le molteplici metriche numeriche, distribuzioni cronologiche e tabelle comparative in sezioni testuali discorsive e rigorose, inserite nel PDF finale. Questi risultati sono di facile comprensione e risultano immediatamente evidenti a chiunque esamini il report.

Grazie a questa architettura integrata, il tempo complessivo di redazione del report tattico pre-gara si è ridotto da diverse ore di elaborazione manuale a un processo semi-automatico, consentendo allo staff di focalizzare le proprie energie cognitive esclusivamente sulla progettazione della strategia di campo.

5.2 FEEDBACK QUALITATIVI E RISCONTRI OPERATIVI DELLO STAFF TECNICO

La validazione empirica e applicativa della piattaforma non può prescindere dal riscontro sul campo. L'infrastruttura è stata sottoposta alla valutazione diretta dell'allenatore e dei collaboratori tecnici del **A.C. Perugia Calcio**, i quali ne hanno testato l'efficacia operativa in occasione della preparazione delle gare di campionato e dei precedenti allenamenti settimanali.

I feedback qualitativi raccolti durante le sessioni di analisi video e di pianificazione settimanale possono essere sistematizzati in tre direttrici principali:

- **Immediatezza e chiarezza prescrittiva del testo generato:** Un elemento di particolare apprezzamento da parte dello staff ha riguardato la struttura formale e linguistica dei report generati dal modulo NLG. L'adozione di un registro rigoroso, focalizzato e privo di ambiguità ha evitato il rischio di dispersione informativa. Indicazioni prescrittive dirette sono state recepite immediatamente come linee guida operative da tradurre in esercitazioni pratiche durante le sedute di allenamento.

- **L'affidabilità predittiva dei trend temporali e cronometrici:** L'introduzione dell'analisi della distribuzione dei gol in slot temporali di 15 minuti ha fornito allo staff una chiave di lettura inedita rispetto alla tradizionale osservazione visiva. Rilevare statisticamente che la formazione avversaria accusasse picchi di vulnerabilità difensiva in finestre orarie ben delimitate ha permesso di impostare una gestione strategica dei ritmi di gara, orientando le scelte sui cambi e sull'intensità del pressing nei momenti cruciali dell'incontro.

- **Limiti percepiti e margini di calibrazione soggettiva:**
  Accanto ai riscontri ampiamente positivi, il confronto con i tecnici ha fatto emergere alcune considerazioni critiche orientate al miglioramento continuo dello strumento. Tali critiche, o suggerimenti, riguardavano la necessità di comprendere il dato relativo alla posizione. In questo specifico scenario lo staff ha evidenziato come il dato statistico puro, come ad esempio la percentuale di possesso palla, debba essere costantemente affiancato dalla comprensione della tattica posizionale avversaria, che nel caso dell'esempio citato in precedenza può riferirsi alla distinzione tra un blocco basso posizionale e una pressione ultra-offensiva a uomo, suggerendo per le future evoluzioni l'inclusione di metriche di interazione spaziale diretta.

5.3 BILANCIO COMPLESSIVO DEI RISULTATI CONSEGUITI E VALIDAZIONE DELL'ARCHITETTURA.

Il bilancio scientifico, metodologico e ingegneristico del progetto di tesi attesta il pieno raggiungimento degli obiettivi prefissati in sede di pianificazione strategica. L'architettura software sviluppata ha dimostrato sperimentalmente che è possibile colmare il divario storico tra la ricerca accademica in Sports Analytics e la gestione operativa quotidiana all'interno di un club professionistico di Lega Pro.

I risultati conseguiti possono essere sintetizzati attraverso tre pilastri fondamentali:

1. **Rigore computazionale e riproducibilità:** L'affidamento dei calcoli deterministici, delle normalizzazioni anagrafiche e delle analisi statistiche a script Python strutturati ha azzerato i margini di errore umano legati alla compilazione manuale dei fogli di calcolo, garantendo la totale tracciabilità e riproducibilità scientifica di ogni indicatore.

2. **Integrazione eterogenea ed efficienza relazionale:** La capacità di armonizzare flussi informativi differenti (dati anagrafici, statistiche di match, eventi di gol e cartellini disciplinari) all'interno di un unico modello relazionale a stella, gestito tramite Power BI, ha confermato la validità dell'approccio ingegneristico basato su moduli interconnessi.

3. **Scalabilità e sostenibilità economica:** A differenza delle soluzioni software proprietarie ad alto costo in dotazione esclusiva ai club di massima serie, la pipeline sviluppata dimostra come sia possibile realizzare un sistema di supporto decisionale di altissimo valore, sfruttando tecnologie open-source, standard di mercato e dataset strutturati, rendendo la metodologia scalabile e replicabile per qualsiasi altra stagione sportiva o categoria.

5.4 LIMITI ATTUALI DELL'ANALISI DATI NEL CALCIO PROFESSIONISTICO

Nonostante i traguardi conseguiti e l'efficacia operativa dimostrata sul campo, l'analisi condotta consente di evidenziare con lucidità accademica i limiti strutturali che caratterizzano l'attuale stato della Sports Analytics applicata alle categorie professionistiche minori (senza prendere in considerazioni quelle semi professionistiche o dilettantistiche):

- **La granularità limitata dei dataset basati su eventi discreti:** I dati tradizionali ricavati dai match descrivono con precisione cosa è accaduto durante la partita ed il soggetto che ha compiuto l'evento, ma offrono una visibilità limitata sulla componente spaziale dei ventidue calciatori in campo.

- **La forte variabilità comportamentale e contestuale in Serie C:** A differenza dei campionati d'élite (Serie A, Premier League, Bundesliga, ...), dove la stabilità tattica dei collettivi è elevata, i campionati di Lega Pro risentono di una maggiore variabilità legata alla tenuta atletica, alla pressione psicologica e alla frequente rotazione delle rose. Tale variabilità può generare un'ulteriore difficoltà nell'acquisizione di una "certezza assoluta".

- **L'assenza di metriche avanzate di tracciamento posizionale nei fogli standard:** Molti indicatori di nuova generazione, come ad esempio gli Expected Goals basati sulle coordinate geometriche del tiro, richiedono dati di tracking ad alta frequenza che esulano dai dataset tabulari tradizionali, richiedendo in futuro un salto tecnologico verso sistemi di acquisizione ottica avanzata.

5.5 PROSPETTIVE ED EVOLUZIONI FUTURE: COMPUTER VISION E DEEP LEARNING NELLO SPORTS ANALYTICS

A superamento dei limiti attuali e in assoluta continuità con l'architettura modulare sviluppata in questo lavoro di tesi, le evoluzioni future del progetto di analisi si orientano verso le frontiere più avanzate della ricerca internazionale:

- **Sistemi di object detection basati su computer vision**: L'integrazione di reti neurali convoluzionali (CNN) e modelli di visione artificiale all'avanguardia, come le architetture YOLO applicate ai flussi video delle partite, permetteranno di estrarre in autonomia le coordinate cartesiane (x,y) dei giocatori e del pallone direttamente dalle riprese televisive o fisse, abbattendo i costi di acquisizione dei dati spaziali e azzerando la dipendenza da provider esterni.

- **Modelli di tracciamento spaziale e pitch control:** La transizione da un'analisi basata su eventi discreti a un'analisi basata su traiettorie continue consentirà di calcolare in tempo reale indici di controllo territoriale dinamici e di superficie libera, offrendo allo staff tecnico delle misure geometriche immediate della densità delle linee di pressione avversarie.

- **Evoluzione del modulo NLG verso modelli multimodali avanzati:** L'impiego futuro di Large Language Models di tipo multimodale, capaci di elaborare simultaneamente tabelle numeriche e sequenze video estratte dalle gare, aprirà la strada a report tattici totalmente immersivi e assistiti da intelligenza artificiale generativa in tempo reale.

5.6 CONSIDERAZIONI CONCLUSIVE

Il percorso di ricerca, progettazione e sviluppo descritto in questa tesi ha dimostrato come l'ingegneria del software e la data science applicata non debbano essere interpretate come entità estranee o accademiche rispetto alla dinamicità del campo da gioco, bensì come strumenti di supporto strategico imprescindibili per il moderno staff tecnico.

La perfetta simbiosi tra la precisione computazionale deterministica di Python, l'esplorazione visiva interattiva di Power BI e la sintesi cognitiva intelligente garantita dalle API di Gemini Pro delinea un paradigma metodologico solido, replicabile ed estremamente efficiente.

Questo lavoro pone basi concrete per una nuova generazione di strumenti di supporto decisionale nel calcio professionistico, dimostrando che il futuro della tattica sportiva risiede nell'armonizzazione rigorosa tra la complessità della scienza dei dati e la pragmatica intelligenza del rettangolo verde.

BIBLIOGRAFIA

[1] https://www.python.org/

[2] https://docs.posit.co/ide/user/#rstudio-ide-oss-downloads

[3] https://www.w3schools.com/html/

[4] https://www.w3schools.com/Css/

[5] https://www.w3schools.com/js/

[6] https://www.djangoproject.com/

[7] https://gemini.google.com/app?hl=it

[8] https://www.microsoft.com/it-it/power-platform/products/power-bi?market=it

[9] https://www.microsoft.com/it-it/microsoft-365/excel

[10] https://www.mysql.com/

[11] https://sqlite.org/index.html

[12] https://pandas.pydata.org/

[13] https://numpy.org/

[14] https://code.visualstudio.com/

[15] https://git-scm.com/

[16] https://github.com/

[17] https://www.football.com/livescore/it

[18] https://www.diretta.it/

[19] https://it.besoccer.com/

[20] https://www.transfermarkt.it/

[21] https://app.diagrams.net/

[22] https://blog.performancelab16.com/acute-chronic-workload/

[23] https://www.scienzemotorie.com/articoli/lacwr-nel-calcio-delite-gestione-del-microciclo-competitivo-biomeccanica-e-il-dibattito-sulla-prevenzione-degli-infortuni

[24] https://www.w3schools.com/html/html5_canvas.asp