

Progetto Ontologia - Calcio

Rosso Lorenzo - Matricola 945995

Anno Accademico 2024/2025

Modellazione Concettuale per il Web Semantico

Contents

1	Motivazioni	3
2	Requirements	4
2.1	Finalità	4
2.2	Task e contesto	4
2.3	Utenti a cui si rivolge	5
3	Descrizione del Dominio	6
3.1	Dominio	6
3.2	Fonti	7
4	Competency Questions	8
5	Documentazione	9
5.1	Casi reali di ispirazione	9
5.1.1	SofaScore	9
5.1.2	Transfermarkt	10
5.2	Allineamento con ontologie esistenti	11
5.2.1	Schema.org/SportsTeam	11
5.2.2	FOAF/Person	11
5.2.3	DC Dublin Core	11
5.2.4	BBC Sport Ontology	11
5.3	Esempio reale nella A-box	12
6	Lode	14
7	Visualizzazione	15
7.1	Tassonomia delle classi	15
7.2	Template utilizzati	15
7.3	Triple dell'esempio in forma di tabella	16
8	Ontologia in formato Turtle	18

9	Query SPARQL	18
9.1	Utente "Match Analyst"	18
9.1.1	Query 1	18
9.1.2	Query 2	19
9.1.3	Query 3	20
9.2	Utente "Tifoso"	21
9.2.1	Query 4	21
9.2.2	Query 5	22
10	Applicazione Client	24
10.1	User Interface	24
10.2	Analisi del Codice	25

1 Motivazioni

Ho deciso di realizzare il presente progetto modellando il dominio dello sport del Calcio.

Tale scelta è stata dettata dal fatto che il Calcio è uno sport culturalmente parecchio diffuso in tutto il mondo, ed inoltre presenta un dominio chiaro e interessante da modellare.

Il calcio, pur essendo di base uno Sport molto semplice, presenta una serie di aspetti nel suo insieme che possono risultare parecchio interessanti da modellare formalmente, tra cui le Squadre, i Giocatori, le Partite, i Ruoli... e di conseguenza tutte le relazioni che derivano tra queste varie entità. Inoltre avere uno storico di dati strutturati in questo ambito è sicuramente prezioso per appassionati e addetti ai lavori.

L'obiettivo dell'ontologia è soprattutto quello di fornire informazioni sul dominio calcistico soprattutto per le nuove figure professionali e business che si stanno diffondendo in tale ambito negli ultimi anni, quali le figure di Match Analyst e Football Data Analyst (vedi ad esempio il caso reale di Opta Stats Perform), per i quali poter reperire informazioni sul mondo calcistico è fondamentale.



2 Requirements

2.1 Finalità

L'ontologia è stata realizzata con lo scopo di supportare l'analisi dei dati relativi al calcio, in particolare focalizzandosi sulle prestazioni (performance) dei giocatori nelle varie partite.

Questa ontologia vuole quindi offrire una rappresentazione chiara dei dati in tale ambito, consentendo di organizzare informazioni complesse in un formato che sia facilmente interpretabile e utilizzabile per diverse applicazioni.

Tra le finalità principali vi è la possibilità di monitorare e valutare parametri fondamentali nella Match Analysis (ossia la disciplina che studia le statistiche nel mondo del calcio), come expected goals (xG), passaggi completati, goal realizzati, assist forniti rilevanti per la performance individuale.

Oltre alla consultazione di questi dati ai fini della Match Analysis, l'ontologia può essere anche un punto di riferimento per la consultazione visuale di tali dati anche ai non esperti, con possibilità di consultare statistiche complesse in modo efficace e chiaro.

2.2 Task e contesto

I task specifici dell'ontologia si concentrano principalmente sulla consultazione di dati relativi al contesto dei giocatori, oltre che i dati statistici che tali giocatori hanno fatto registrare durante le partite.

In particolare, i task principali che vuole ricoprire un'ipotetica applicazione sono i seguenti:

- Consultazione di dati generali sui giocatori: indicazione del ruolo di ogni giocatore e squadra di appartenenza.
- Accesso alla performance di specifici giocatori nelle varie partite: un dato del genere è fondamentale per monitorare l'andamento nel tempo delle prestazioni dei giocatori.
- Dati sul numero di partite disputate dai giocatori, per confrontare dati sull'impiego e per poterli comparare con le performance registrate.

Il contesto principale in cui si colloca l'ontologia è quello della match data analysis, dove la necessità di estrarre insight precisi e azionabili dai dati calcistici è fondamentale per migliorare le performance e l'efficienza in tale ambito.

2.3 Utenti a cui si rivolge

L'ontologia è pensata per soddisfare le esigenze di diversi tipi di utenti.

- Gli analisti sportivi possono utilizzarla per studiare le prestazioni dei giocatori e delle squadre, individuando trend e fornendo supporto alle strategie tattiche.
- Appassionati di calcio che possono trarre beneficio dall'ontologia, approfondendo la propria conoscenza sul mondo del calcio a livello statistico.

3 Descrizione del Dominio

3.1 Dominio

Il dominio rappresenta l'ambito del gioco del calcio, e si concentra su vari aspetti chiave del gioco quali le partite, i giocatori, le squadre, le loro posizioni in campo e, soprattutto, le performance che si possono misurare durante un incontro.

L'ontologia che ho realizzato si basa sulle seguenti classi principali:

- Match: rappresenta una partita di calcio, suddivisa in due categorie:
 - InternationalMatch, che include partite tra squadre di nazionali (NationalTeam).
 - LocalMatch, per rappresentare partite tra squadre di club (ClubTeam).
- Team: identifica una squadra di calcio, sia essa una squadra di club (ClubTeam, ad esempio il *Real Madrid*) o una squadra nazionale (NationalTeam, ad esempio *Francia*). Le squadre possono essere coinvolte in partite a livello locale o internazionale.
- Player: rappresenta i giocatori di calcio, con la variante dei NationalPlayer che identificano giocatori che giocano per la loro Nazionale.
- Position: descrive il ruolo di un giocatore in campo. E' stata modellata ogni possibile posizione (portiere, difensore, centrocampista, attaccante).
- Performance: tiene traccia delle statistiche individuali dei giocatori in ogni partita, cioè quante reti sono state segnate, quanti passaggi completati, quanti gol attesi (XG), quanti assist realizzati. Questo è il cuore della mia ontologia, pensata per analizzare il contributo specifico di ciascun giocatore.

La struttura permette di rappresentare sia la complessità del calcio, con statistiche dettagliate dei giocatori per ogni partita, sia le relazioni più semplici tra squadre, giocatori e match. L'obiettivo è offrire una base utile per modellare l'analisi di dati calcistici, applicabile sia a scenari professionistici (per le figure che lavorano in tale ambito) che per gli appassionati o giornalisti.

3.2 Fonti

Per costruire questo modello, mi sono affidato a diverse fonti online e materiali di riferimento, oltre alla mia conoscenza personale del calcio:

- Per le statistiche e le performance dei giocatori:
 - <https://www.sofascore.com/>, utile per riuscire a pescare dati e statistiche sui giocatori nell'archivio delle partite passate, da cui ho preso spunto ad esempio per la *performance di Mbappè nella Finale Mondiale 2022*, modellata nell'ontologia come individuo.
 - <https://www.statsperform.com/opta/>, un sito di riferimento nell'analisi statistica calcistica.
- Per le squadre di appartenenza e le posizioni dei giocatori in campo:
 - <https://www.transfermarkt.com/>, un sito completo che contiene il database di ogni giocatore in attività e non, con relativi dati sulla squadra di appartenenza e la posizione in campo.
 - Conoscenze personali del gioco.
- Per le partite:
 - Wikipedia, molto utile per avere dati precisi sulle partite più importanti svolte nel passato. Vedi https://it.wikipedia.org/wiki/Finale_del_campionato_mondiale_di_calcio_2022 da cui ho preso spunto.
- Per la modellazione dell'ontologia:
 - Mi sono basato sull'utilizzo e sulla conoscenza accademica acquisita durante il corso del Software Protégé, per strutturare classi e relazioni in modo formale.

4 Competency Questions

Le domande tipo per mostrare l'utilità della mia ontologia, che verranno analizzate in modo approfondito e tradotte sotto forma di query SPARQL nei seguenti capitoli, sono le seguenti.

- **Quanti gol ed expected goals ha totalizzato un determinato giocatore in una determinata partita?**

Spiegazione: Conta il numero di gol (`nGoals`) e di expected goals (`nExpectedGoals`) realizzati da un giocatore in una partita specifica.

Utilità: permette il confronto tra possibili occasioni da rete e occasioni realizzate, indice per monitorare l'abilità di finalizzazione del giocatore.

- **Quale partite ha giocato uno specifico giocatore X?**

Spiegazione: Elenca le partite a cui il giocatore X ha partecipato. *Utilità:* permette di verificare quali partite ha giocato ogni giocatore di una squadra.

- **Quanti giocatori hanno un dato X superiore al giocatore Y nella partita Z?**

Spiegazione: Somma i giocatori che presentano un valore maggiore in una determinata data property registrata in una specifica partita.

Utilità: permette di confrontare la bontà della prestazione del giocatore Y.

- **Elencami i giocatori e la loro relativa squadra di Club e ruolo**

Spiegazione: Restituisce l'elenco dei giocatori di club con la squadra a cui appartengono e il ruolo che ricoprono in campo.

Utilità: permette di accedere a dati quali ruolo e squadra di ogni giocatore, utile a fini di consultazione o curiosità.

- **Elencami i giocatori che sono convocati in nazionale**

Spiegazione: Restituisce un elenco di tutti i giocatori che giocano in squadre nazionali.

Utilità: permette di consultare le selezioni dei Commissari Tecnici delle proprie nazionali del cuore.

5 Documentazione

5.1 Casi reali di ispirazione

Ho preso ispirazione da diverse fonti online nell'ambito calcistico per riuscire a progettare l'ontologia al meglio. Tra le principali risorse considerate ho utilizzato:

5.1.1 SofaScore

SofaScore è una piattaforma che fornisce dati dettagliati sulle performance dei giocatori durante le partite. Le statistiche offerte includono:

- Gol segnati
- Expected Goals (xG)
- Assist e passaggi
- Altri dati interessanti che però la presente ontologia non si pone come obiettivo, quali le heatmap o i dati sulle posizioni medie in campo dei vari giocatori.

Ad esempio, nella finale della Coppa del Mondo FIFA 2022 tra Francia e Argentina, Lionel Messi ha fornito una grande performance, segnando una doppietta e realizzando 85 tocchi palla. L'immagine sottostante mostra un esempio della schermata di SofaScore relativa alla sua prestazione.

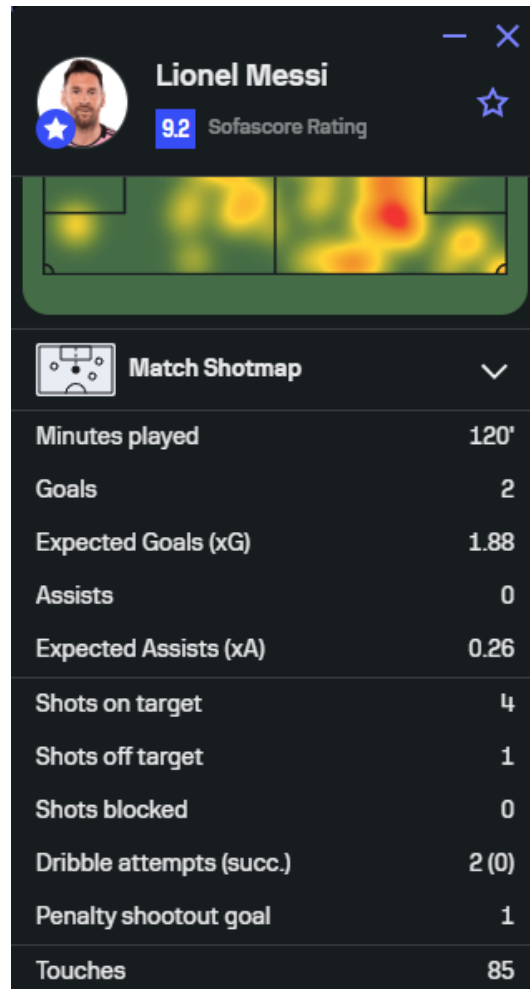


Figure 1: Dati di SofaScore sulla prestazione di Lionel Messi nella finale Francia-Argentina.

5.1.2 Transfermarkt

Transfermarkt è una piattaforma web specializzata in dati relativi ai giocatori, alle squadre e al mercato calcistico. Tra le informazioni fornite ci sono:

- Appartenenza attuale di un giocatore a una squadra.
- Dati anagrafici del giocatore.
- Ruolo e posizione in campo del giocatore.
- Altri dati irrilevanti per l'attuale ontologia, quali il valore di mercato o lo storico delle squadre passate.

Ad esempio, la figura sottostante mostra una schermata di Transfermarkt che evidenzia i dati anagrafici, il ruolo e la squadra di Jude Bellingham.

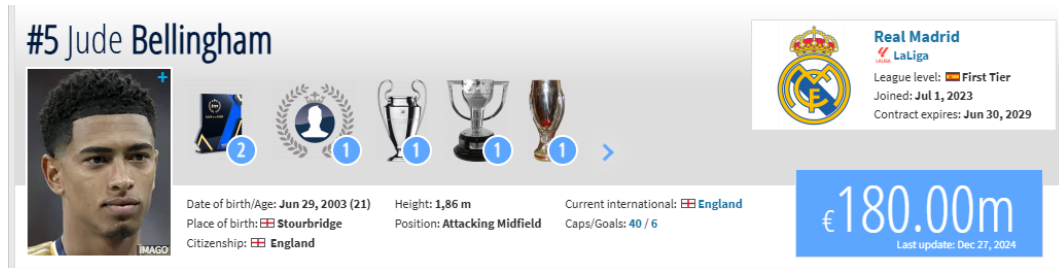


Figure 2: Dati di Transfermarkt su Jude Bellingham.

5.2 Allineamento con ontologie esistenti

Per garantire un'adeguata interoperabilità e conformità agli standard, l'ontologia proposta è stata allineata con le seguenti ontologie:

5.2.1 Schema.org/SportsTeam

Ho utilizzato Schema.org/SportsTeam per allineare la mia classe Team, relativa alle squadre. Questa ontologia consente di modellare informazioni strutturate come il nome della squadra, i membri e le competizioni partecipate.

5.2.2 FOAF/Person

FOAF (Friend of a Friend) è stata adottata per rappresentare i giocatori come individui di sottoclasse Person, allineata appunto tramite l'ontologia FOAF.

5.2.3 DC Dublin Core

Le annotazioni relative al progetto, come titolo, data e creatore, sono state modellate utilizzando lo standard Dublin Core. Questo standard è ampiamente riconosciuto per la descrizione di risorse e metadati.

5.2.4 BBC Sport Ontology

Per allineare la classe Match ho fatto delle ricerche sul repository di vocabolari BBC, e in particolare l'ontologia Sport Ontology ha fatto al caso mio, e ho provveduto ad allineare la classe Match della mia ontologia.

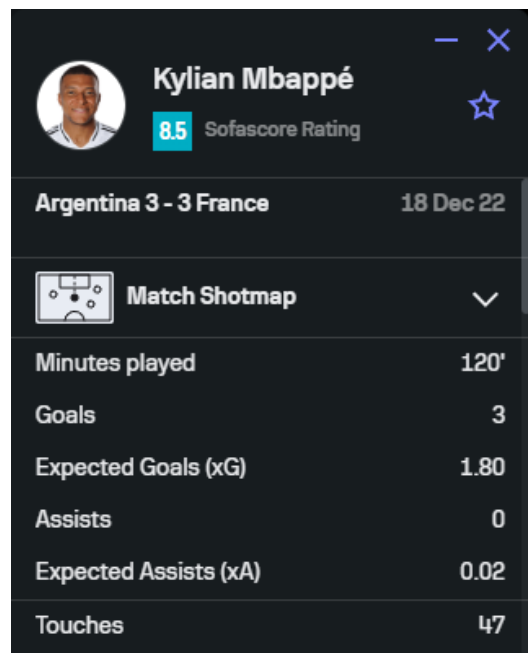
5.3 Esempio reale nella A-box

Illustrerò ora di seguito le schermate delle fonti che ho utilizzato per estrapolare, e quindi compilare, i dati A-Box della mia ontologia, sfruttando i portali SofaScore e Transfermarkt come indicato nella sezione precedente.

Match: Francia vs Argentina

Performance di riferimento:

- Giocatore: *Kylian Mbappé*.
- Gol segnati: *3*.
- Expected Goals (xG): *1.8*.
- Assist: *0*.
- Touches: *47*.



The screenshot shows a player profile for Kylian Mbappé with a Sofascore Rating of 8.5. Below this, it lists the match as Argentina 3 - 3 France on 18 Dec 22. A 'Match Shotmap' icon is visible. The statistics table below shows the following data:

Statistic	Value
Minutes played	120'
Goals	3
Expected Goals (xG)	1.80
Assists	0
Expected Assists (xA)	0.02
Touches	47

Figure 3: Dati SofaScore sulla Performance di Kylian Mbappé in Francia Argentina finale mondiale 2022

Successivamente, per riempire i dati relativi alla squadra di appartenenza e al ruolo, mi sono servito della piattaforma TransferMarket, di cui indico di seguito la schermata da cui ho prelevato i seguenti dati.

- Nominativo: *Kylian Mbappé*.
- Ruolo: *Forward (attaccante)*.
- Età: *26*.
- Squadra: *Real Madrid*.

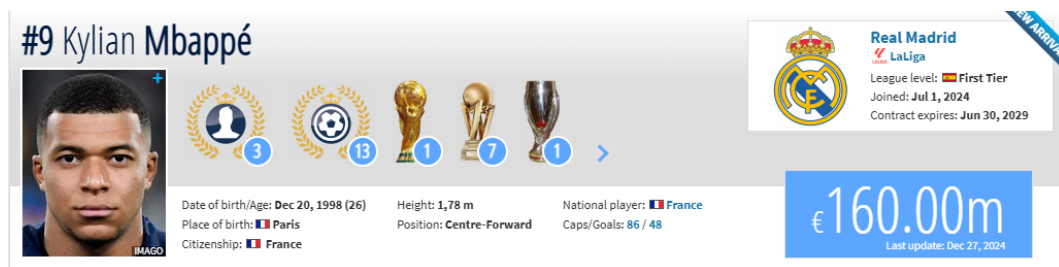


Figure 4: Dati TransferMarket di Kylian Mbappè

6 Lode

Ho deciso di generare la documentazione della presente ontologia tramite il tool LODE.

E' possibile visualizzare e analizzare quest'ultima al seguente link:
documentazione LODE.

7 Visualizzazione

7.1 Tassonomia delle classi

Di seguito allego la tassonomia visuale delle classi incluse nella mia ontologia. E' stata generata tramite il plugin OntoGraf aggiunto al software Protegè.

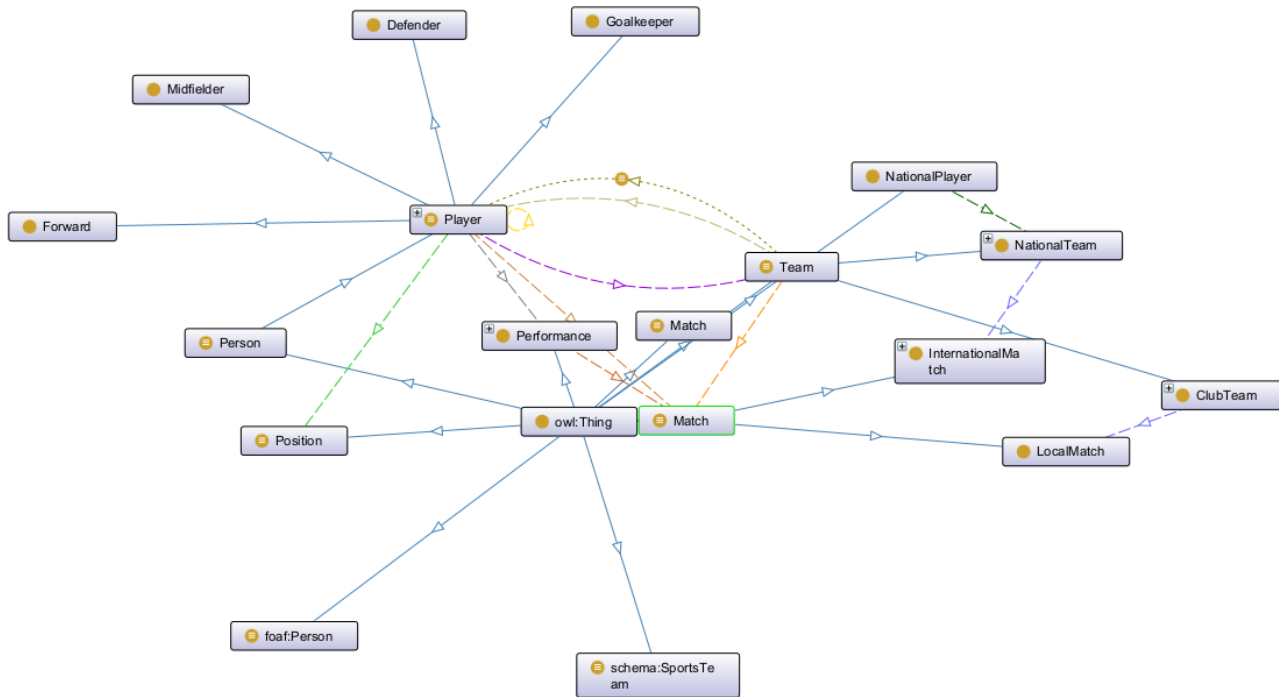


Figure 5: Tassonomia delle classi ontologia Football

7.2 Template utilizzati

Per la modellazione delle posizioni (*Position*) ho utilizzato il pattern AgentRole.

In particolare come si vede nella tassonomia, la classe *Player* è in relazione tramite la object property *hasPosition* con la classe *Position*. Tramite questo pattern è anche possibile fare affermazioni sui ruoli senza coinvolgere direttamente gli agenti che giocano quei ruoli.

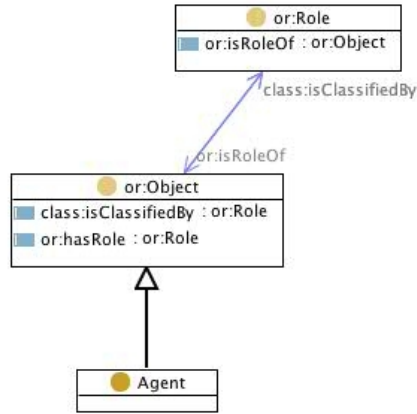


Figure 6: Pattern AgentRole

7.3 Triple dell'esempio in forma di tabella

Di seguito mostrerò l'insieme delle triple che rappresentano la richiesta di informazioni relative al giocatore *Jude Bellingham*, esattamente come nell'esempio di TransferMarket illustrato nella sezione 5.1.2.

	predicate	↕	object
1	rdf.type		http://www.semanticweb.org/football#Player
2	rdf.type		http://www.semanticweb.org/football#Person
3	rdf.type		http://xmlns.com/foaf/0.1/Person
4	rdf.type		_node6
5	rdf.type		owl:NamedIndividual
6	rdfs:comment		"Professional football player"@en
7	rdfs:label		"JudeBellingham"@en
8	http://www.semanticweb.org/football#playsFor		http://www.semanticweb.org/football#RealMadrid
9	http://www.semanticweb.org/football#hasPosition		http://www.semanticweb.org/football#Midfielder
10	http://www.semanticweb.org/football#isTeammateOf		http://www.semanticweb.org/football#KylianMbappe
11	http://www.semanticweb.org/football#isTeammateOf		http://www.semanticweb.org/football#JudeBellingham
12	http://www.semanticweb.org/football#age		"21"^^xsd:integer
13	http://www.semanticweb.org/football#name		"Jude Bellingham"

Figure 7: Triple dell'esempio Jude Bellingham

Inoltre allego di seguito il Visual Graph relativo all'esempio reale sopra citato

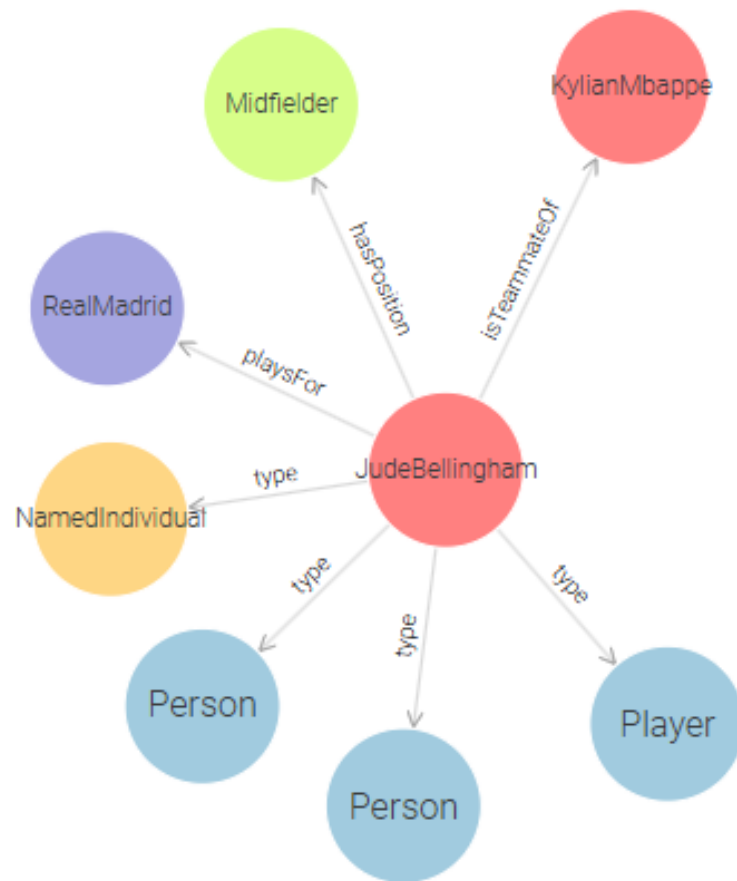


Figure 8: Visual Graph dell'esempio reale Jude Bellingham

8 Ontologia in formato Turtle

Al seguente link è possibile raggiungere la mia ontologia esportata in formato Turtle, compresa di tutte le indicazioni fornite nelle specifiche del progetto.

Versione senza inferenze

Versione con inferenze materializzate - Reasoner HermiT 1.4

Versione con inferenze materializzate - Reasoner Pellet

Ho testato la mia ontologia con due reasoner diversi, cioè HermiT 1.4 e Pellet.

Ho notato la minor occupazione del file ottenuto dall'esportazione inferenze tramite Pellet, che pare quindi più veloce e leggero.

9 Query SPARQL

Di seguito illustrerò 5 query SPARQL che andranno a rispondere a dei possibili task che richiedono l'interrogazione della presente ontologia.

Nel dettaglio verranno presentate 3 casistiche che riguardano utenti del tipo "match analysts", e quindi figure professionali che vogliono ottenere dati statistici a fini di ricerca, mentre le restanti 2 casistiche riguardano l'utente "tifoso", che pone le sue query a fine di curiosità o approfondimento personale.

9.1 Utente "Match Analyst"

9.1.1 Query 1

Il flusso dell'utente Match Analyst inizia con la richiesta dell'elenco delle partite giocate da uno specifico giocatore (nella mia ontologia per semplicità ho preso in considerazione la sola stagione 2022), in modo da poter scegliere su quale concentrare la propria analisi del giocatore. In questo caso è stato richiesto il giocatore "Kylian Mbappè".

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX Football: <http://www.semanticweb.org/football#>
```

```
SELECT ?match
```

```
WHERE {
```

```
    ?player Football:name "Kylian Mbappè" .
```

```
    ?player Football:participatesIn ?match .
```

```
}
```

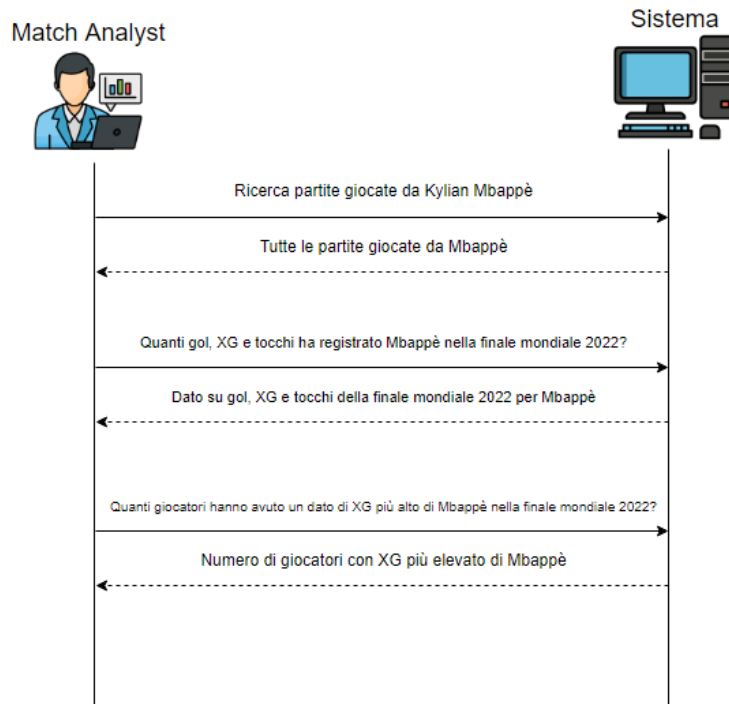


Figure 9: Flusso dell'utente "Match Analyst"

<i>?match</i>
Football:WorldCupFinal2022

Risultati della query:

La query ritorna le partite giocate da "Kylian Mbappè" lasciando come elemento variabile *?match*.

NB: Viene ritornato un solo risultato poichè per semplicità ho creato nella mia ontologia il solo esempio di partita giocata della finale mondiale 2022 per Mbappè.

9.1.2 Query 2

Il flusso continua con la richiesta da parte del Match Analyst dei dati di Mbappè relativi ad una delle partite in cui ha giocato, in questo caso la finale dei mondiali 2022.

Poniamo caso che stia indagando sul rapporto tra gol segnati ed attesi, oltre che dai tocchi palla per analizzare la partecipazione al gioco del giocatore in questione durante la partita su cui si è focalizzato.

PREFIX rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>

PREFIX Football: <http://www.semanticweb.org/football#>

```
SELECT ?goals ?xG ?touches
WHERE {
  ?player Football:name "Kylian Mbappè" .
  ?player Football:hasPerformance ?performance .
  ?performance Football:referredTo Football:WorldCupFinal2022 .
  ?performance Football:nGoals ?goals .
  ?performance Football:nExpectedGoals ?xG .
  ?performance Football:nTouches ?touches .
}
```

<i>?goals</i> 1	<i>?xG</i>	<i>?touches</i>
3	1.8	47

La query prima di tutto trova il giocatore chiamato "Kylian Mbappé", recupera la sua performance collegata alla partita "WorldCupFinal2022", e infine ne estrae i dati su nGoals, nExpectedGoals e nTouches. Risultati della query:

9.1.3 Query 3

Per concludere il caso tipo, l'utente Match Analyst vuole capire **quanti** giocatori che hanno partecipato alla finale Mondiale 2022 hanno avuto un dato di Expected Goals xG superiore a quello di Kylian Mbappé, che ha siglato una tripletta.

Sorprendentemente risulta 1 giocatore con xG superiore.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX Football: <http://www.semanticweb.org/football#>
```

```
SELECT (COUNT(?giocatore) AS ?numeroGiocatori) WHERE {
  # Prelevo il valore di xG di Mbappé nella finale
  Football:KylianMbappe Football:hasPerformance ?mbappePerformance .
  ?mbappePerformance Football:referredTo Football:WorldCupFinal2022 ;
  Football:nExpectedGoals ?xgMbappe .

  # Pesco tutti i giocatori che hanno giocato la finale e il loro xG
  ?giocatore Football:hasPerformance ?performance .
  ?performance Football:referredTo Football:WorldCupFinal2022 ;
  Football:nExpectedGoals ?xgGiocatore .

  # Filtro i giocatori con xG maggiore di Mbappé
```

```

    FILTER(?xgGiocatore > ?xgMbappe)
}

```

Risultati della query:

<i>?numeroGiocatori</i>
1

La query SPARQL appena illustrata è virtualmente divisa in 3 sezioni, poichè nella prima parte viene prelevato il valore di xG di Mbappè nella finale; successivamente vengono pescati tutti i giocatori che hanno giocato la finale e il loro rispettivo xG; infine vengono filtrati i giocatori con xG maggiore di quello di Mbappè, che contribuiranno poi al conteggio dichiarato dal *COUNT* in prima riga.

9.2 Utente "Tifoso"



Figure 10: Flusso dell'utente "Tifoso"

9.2.1 Query 4

Il flusso dell'utente tipo "Tifoso", inizia con la sua volontà di conoscere i giocatori di Club e il loro ruolo, oltre che la loro squadra di Club di appartenenza.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX Football: <http://www.semanticweb.org/football#>

SELECT ?giocatore ?squadra ?posizione WHERE {
    ?giocatore a Football:Player ;

```

```

    Football:playsFor ?squadra;
    Football:hasPosition ?posizione .
    ?squadra a Football:ClubTeam
}

```

Risultati della query:

giocatore	squadra	posizione
Football:LionelMessi	Football:InterMiami	Football:Forward
Football:KylianMbappe	Football:RealMadrid	Football:Forward
Football:JudeBellingham	Football:RealMadrid	Football:Midfielder

La query recupera tutti i giocatori di calcio insieme alla loro squadra e alla loro posizione in campo, assicurandosi che la squadra sia un club e non una nazionale. Prima identifica gli individui che appartengono alla classe `Football:Player`, poi cerca la squadra per cui giocano attraverso la proprietà `Football:playsFor` e la loro posizione tramite `Football:hasPosition`. Infine, aggiunge un filtro che garantisce che la squadra appartenga alla classe `Football:ClubTeam`, escludendo così eventuali squadre nazionali.

9.2.2 Query 5

Infine l'utente tifoso decide per sua curiosità di leggere le convocazioni in Nazionale dei vari commissari tecnici, visionando tutti i giocatori nazionali con relativa squadra di appartenenza.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX Football: <http://www.semanticweb.org/football#>

SELECT ?giocatore ?nazionale WHERE {
    ?giocatore a Football:Player ;
    Football:playsFor ?nazionale .

    ?nazionale a Football:NationalTeam .
}

```

Risultati della query:

giocatore	nazionale
Football:LionelMessi	Football:ArgentinaNationalTeam
Football:CristianoRonaldo	Football:PortugalNationalTeam
Football:KylianMbappè	Football:FranceNationalTeam

Questa query recupera i giocatori che sono convocati in una nazionale e la nazionale in cui giocano. Trova tutti i giocatori che appartengono alla classe `Football:Player` e sono associati a una squadra tramite `Football:playsFor`. Mi assicuro poi che la squadra appartenga alla classe `Football:NationalTeam`, escludendo così i club e recuperando solo i giocatori delle nazionali.

10 Applicazione Client

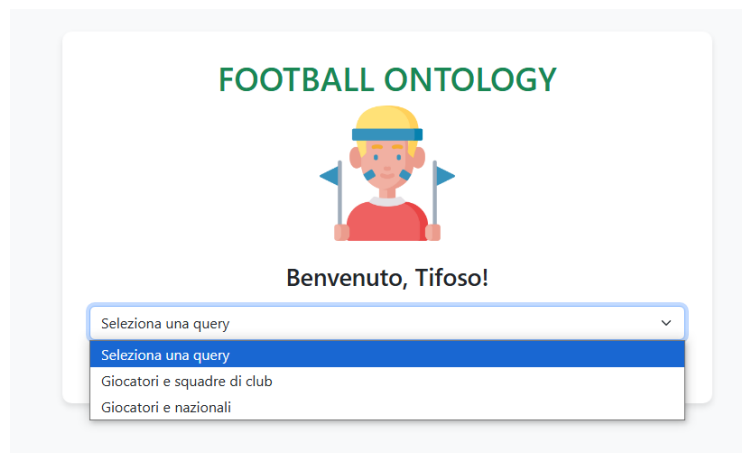
10.1 User Interface

L'applicazione Client è stata implementata utilizzando le tecnologie HTML, CSS compreso del framework Bootstrap, e JavaScript con relativo framework JQuery.

L'interfaccia è stata volutamente realizzata in modo semplice e pulito, e si presenta all'apertura con un menù iniziale di scrematura che prevede due pulsanti iniziali che fungono da Gateway per selezionare il tipo di utente che vuole utilizzare la piattaforma.



Nel momento in cui viene cliccata una delle due voci, l'utente verrà riportato alla schermata ad egli dedicata, in cui si troverà davanti una tendina dalla quale potrà selezionare la domanda che vuole porre al sistema.



Una volta scelta una voce dalla tendina proposta, l'utente visualizzerà immediatamente a schermo la tabella riepilogativa contenente i risultati estratti dalla query.



10.2 Analisi del Codice

Il codice dietro l'applicazione presenta fondamentalmente due punti salienti:

- La composizione della query basata sulla scelta a tendina dell'utente. Di seguito l'esempio dell'utenza Match Analyst.
E' possibile visualizzare tale codice nella figura 11 nella pagina seguente.
- La richiesta a GraphDB di eseguire la query SPARQL e di ritornare i dati. Per farlo mi sono servito della tecnologia AJAX.
E' possibile visualizzare tale codice nella figura 12 nella pagina seguente.

```

if (selectedQuery == 1) {
    //Query che ritorna tutte le partite giocate da Mbappè
    query = `PREFIX Football: <http://www.semanticweb.org/football#>
            SELECT ?match WHERE {
                ?player Football:name "Kylian Mbappè".
                ?player Football:participatesIn ?match. }`;
} else if (selectedQuery == 2) {
    //Query che ritorna le statistiche della prestazione di Mbappè in finale
    query = `PREFIX Football: <http://www.semanticweb.org/football#>
            SELECT ?goals ?xG ?touches WHERE {
                ?player Football:name "Kylian Mbappè".
                ?player Football:hasPerformance ?performance.
                ?performance Football:referredTo Football:WorldCupFinal2022.
                ?performance Football:nGoals ?goals.
                ?performance Football:nExpectedGoals ?xG.
                ?performance Football:nTouches ?touches. }`;
} else if (selectedQuery == 3) {
    //Query che ritorna il numero di giocatori che hanno avuto xG più alto di Mbappè in finale
    query = `PREFIX Football: <http://www.semanticweb.org/football#>
            SELECT (COUNT(?giocatore) AS ?numeroGiocatori) WHERE {
                Football:KylianMbappe Football:hasPerformance ?mbappePerformance.
                ?mbappePerformance Football:referredTo Football:WorldCupFinal2022;
                Football:nExpectedGoals ?xgMbappe.
                ?giocatore Football:hasPerformance ?performance.
                ?performance Football:referredTo Football:WorldCupFinal2022;
                Football:nExpectedGoals ?xgGiocatore.
                FILTER(?xgGiocatore > ?xgMbappe) }`;
} else {
    return;
}

```

Figure 11: Definizione query SPARQL

```

// Esecuzione query tramite AJAX, la risposta mi arriverà in formato textplain
$.ajax({
  url: endpoint,
  type: "GET",
  dataType: "text",
  data: { query: query },
  success: function(data) {
    console.log(data);

    //Formatto la risposta e rimuovo righe vuote
    var results = data.split('\n').filter(function(row) {
      return row.trim() !== ""; // Tolgo le righe vuote
    });

    if (results.length > 0) {
      var theadRow = $("<tr>");

      // Creo intestazione colonna
      var headers = results[0].split(",");
      headers.forEach(header => {
        theadRow.append("<th>" + removePrefix(header) + "</th>");
      });
      $("#resultsTable thead").append(theadRow);

      // Dati tabella con risultati query, evito il primo con .slice poiché è intestazione
      results.slice(1).forEach(row => {
        var tr = $("<tr>");
        var cells = row.split(",");
        cells.forEach(cell => {
          tr.append("<td>" + removePrefix(cell) + "</td>");
        });
        $("#resultsTable tbody").append(tr);
      });
    } else {
      $("#resultsTable tbody").append("<tr><td colspan='3'>Nessun risultato trovato</td></tr>");
    }
  },
  error: function(error) {
    console.error("Errore nella richiesta SPARQL", error);
  }
});

```

Figure 12: Chiamata AJAX verso GraphDB