



Título: WS - Assignment 2

Autores: Gonçalo Silva 103668, Guilherme Antunes 103600, Pedro Rasinhas 103541

Data: 07/06/2024

1. Introdução.....	2
2. Definição da Ontologia.....	2
3. Inferências com SPIN.....	5
4. Novas operações sobre os dados - SPARQL.....	6
5. Integração de dados externos.....	8
6. Publicação de Dados Semânticos.....	12
7. Funcionalidades da Aplicação.....	14
1. Listar jogadores e ordená-los/filtrá-los por diversas opções.....	15
2. Novos Detalhes de Equipa.....	16
3. Aceder e editar squads criados pelo utilizador.....	17
8. Executar Aplicação.....	18
9. Trabalho Futuro.....	19
10. Conclusão.....	19
Referências.....	20

1. Introdução

Este trabalho foi desenvolvido no âmbito da Unidade Curricular de Web Semântica e visa o desenvolvimento de um sistema de informação baseado na Web, dando continuação ao projeto desenvolvido anteriormente nesta cadeira.

O tema escolhido para explorar os objetivos propostos para o projeto foi o videojogo FIFA 24, oficialmente conhecido como EA SPORTS FC24, mais especificamente as cartas dos jogadores do mesmo. Cada carta corresponde a um jogador real e aos atributos jogáveis que o mesmo tem dentro do jogo. Cada jogador tem associado a si, para além dos atributos pessoais, um clube, uma nacionalidade, um género, uma posição, etc.

A implementação descrita neste relatório procura explorar as capacidades de uma ontologia construída com RDFS e OWL, inferências SPIN, queries SPARQL, integração de dados de fontes externas, no caso da Wikidata e a publicação de dados semânticos em RDFa.

Para a realização deste projeto foi usada uma combinação de tecnologias tais como HTML, CSS e JavaScript para o frontend, Django para o backend, RDF N3 como estrutura dos dados, ontologia RDFS e OWL para descrever as relações entre estes, GraphDB como base de dados e SPARQL como query language usada para fazer pesquisas sobre os dados.

2. Definição da Ontologia

À semelhança do primeiro projeto os dados obtidos relativos às cartas do FIFA 24 foram obtidos através de uma API [2], que disponibiliza os dados pretendidos, e dois websites que serviram de guia para o display dessa informação [1][3]. Por sua vez as ligas e seus logótipos foram retirados manualmente da mesma fonte de dados que foi utilizada como orientadora de display[3].

Na pasta *dataset* é possível encontrar o script *fifa.py* utilizado para recuperar todos os dados pretendidos, assim como a construção da estrutura RDF N3. Para além dos dados recuperados da API[2]. A opção pelo RDF N3 deveu-se a ser uma representação relativamente mais fácil para a compreensão humana do que a RDF XML e por conseguir condensar muito mais informação que a N-Triples. Na figura abaixo podemos observar o diagrama da estrutura dos dados.

Para modelar e definir as classes e relações entre os dados utilizámos uma ontologia OWL/RDFS que pode ser encontrada em *dataset/ontology.n3* e através das quais definimos as seguintes classes:

Classe	Características
Player	subclasse de foaf:Person
Team	subclasse de foaf:Group e foaf:Organization
League	subclasse de foaf:Organization
Nationality	

Position	
Gender	
Male	subclasse de Gender e Player disjunta de Female
Female	subclasse de Gender e Player disjunta de Male
Squad	subclasse de foaf:Group
SquadPlayer	subclasse de Player

Para além das propriedades naturais inerentes a cada classe que podem ser visualizadas na Figura 1 temos a referir algumas propriedades e restrições que ultrapassam a simplicidade das que estão na figura.

```
fifaplg:Player a rdfs:Class ;
    rdfs:label "Player"@en ;
    rdfs:subClassOf [ a owl:Restriction ;
        owl:onProperty fifaplp:gender ;
        owl:maxCardinality "1"^^xsd:int
    ] .
```

Na classe *Player*, assim como nas classes *Team* e *League* foi definido que apenas pode existir uma propriedade do tipo *fifaplp:gender*, isto serve para impedir que um jogador, equipa e liga possam ser registados com ambos os géneros e criem conflitos nas queries do sistema.

```
fifasqg:Squad a rdfs:Class ;
    rdfs:label "Squad"@en ;
    owl:equivalentClass [ a owl:Restriction ;
        owl:onProperty fifasqp:player ;
        owl:someValuesFrom fifaspg:SquadPlayer ;
        owl:onClass fifasqg:Squad
    ] .
```

Nas classes *Squad* e *SquadPlayer* foram definidos que para uma instância existir é necessário existir pelo menos um *fifaspg:SquadPlayer*, no caso de *Squad*, e um *fifaplg:Player*, no caso de *SquadPlayer*, para que cada instância seja definida.

```
fifagg:Male a owl:Class ;
    rdfs:label "Men's Football"@en ;
    fifagg:shortLabel "Man"@en ;
    rdfs:subClassOf fifagg:Gender ;
    owl:equivalentClass [ a owl:Restriction ;
        owl:onProperty fifaplp:gender ;
        owl:hasValue fifagg:0
    ] .
```

fifagg:Female owl:disjointWith fifagg:Male .

Nas classes *Male* e *Female* foram associados labels próprios uma vez que se pretende a clara identificação das instâncias deste tipo. Para além disso, é definido que devem ser associadas a esta classe todas as instâncias de classes com propriedade *fifaplp:gender* e valor 0, no caso de *Male*, e 1, no caso de *Female*. Esta restrição acabará por inferir que os elementos desta

classe são *Players*, uma vez que apenas eles têm o predicado *fifaplp*. Para evitar conflitos de géneros, definimos que as classes *Male* e *Female* são disjuntas.

```

fifarel:teammate a owl:ObjectProperty ;
    rdfs:label "teammate"@en ;
    a owl:SymmetricProperty ;
    rdfs:domain fifaplg:Player ;
    rdfs:range fifaplg:Player .

```

```

fifaplp:playsFor a owl:ObjectProperty ;
    rdfs:label "playsFor"@en ;
    owl:equivalentProperty fifaplp:team .

```

fifalp:gender que não constam no esquema inicial dos dados e a propriedade *fifapl:playsFor* que é uma propriedade equivalente(sinónima) da propriedade *fifapl:team*.

```

fifaplp:preferredFoot a owl:DataProperty ;
    rdfs:label "preferredFoot"@en ;
    rdfs:domain fifaplg:Player ;
    rdfs:range [
        a rdfs:Datatype ;
        rdfs:label "int"@en ;
        owl:onDatatype xsd:int ;
        owl:oneOf ( "1"^^xsd:int "2"^^xsd:int )
    ] .

```

limitando os valores. Por exemplo, o pé preferido só pode tomar 2 valores distintos definidos pois não é possível ter mais do que 2 pés.

```

fifaplp:skillMoves a owl:DataProperty ;
    rdfs:label "skillMoves"@en ;
    rdfs:domain fifaplg:Player ;
    rdfs:range [
        a rdfs:Datatype ;
        rdfs:label "int"@en ;
        owl:onDatatype xsd:int ;
        owl:withRestrictions (
            [ xsd:minInclusive "1"^^xsd:int ]
            [ xsd:maxInclusive "5"^^xsd:int ]
        )
    ] .

```

Squad apenas pode ter 11 jogadores, mais do que isso criaria conflitos de informação.

A Figura 1 representa a forma como os dados estão organizados no sistema. O esquema da base de dados manteve-se o mesmo do projeto anterior pelo que as características e pormenores se mantêm os mesmos que foram referenciados anteriormente.

Em termos de propriedades de objetos foram criadas algumas novas propriedades que não estavam previstas no esquema original dos dados. Criámos uma propriedade simétrica *fifarel:teammate* que permitirá definir que quaisquer 2 jogadores da mesma equipa são colegas de equipa. Essa inferência será falada mais à frente no SPIN. Também foram criadas novas propriedades como a *fifatp:gender* e a

Nas propriedades *fifapl:preferredFoot*, *fifapl:attackingWorkRate* e *fifapl:defensiveWorkRate* foram definidos os valores específicos que podem tomar. Uma vez que são valores que irão, na maioria dos casos, ser transformados em outras representações gráficas, funcionam como enums, propositadamente

Para finalizar em propriedades como *fifapl:skillMoves*, *fifapl:weakFootAbility*, *fifapl:overallRating* e *fifaspp:position* foram definidos ranges de valores entre os quais pode estar o da propriedade. Os 3 primeiros prendem-se diretamente com as limitações do jogo a que o nosso sistema está associado, sendo que o rating só pode ir de 1 até 99 e as skillMoves e a weakFootability de 1 até 5. A propriedade *fifaspp:position* é limitada de 1 a 11 para garantir que cada

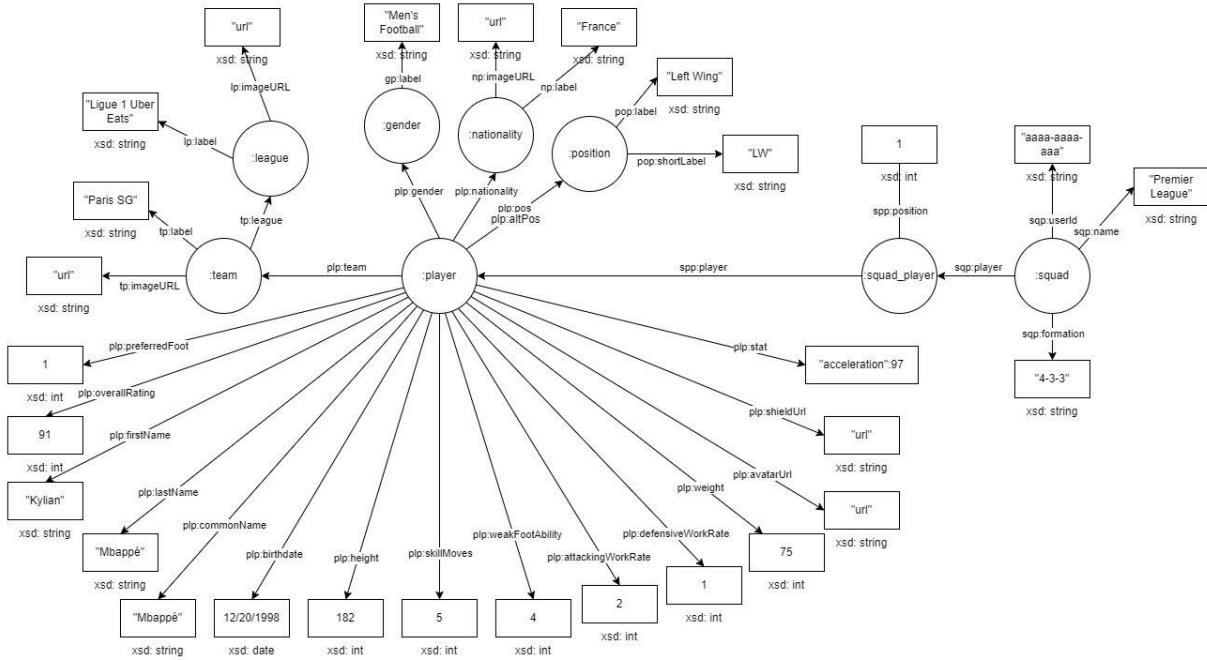


Figura 1 - Esquema da Base de Dados

3. Inferências com SPIN

Nem todas as inferências podem ser feitas apenas definindo a ontologia, para isso criámos 2 queries em notação SPIN para inferir 4 novas relações. Foram elas:

- Teammates

```
inference队友 = """
PREFIX fifaplp: <http://fifa24/player/pred/>
PREFIX fifaplg: <http://fifa24/player/guid/>
PREFIX fifarel: <http://fifa24/relationship/>

INSERT {
    ?player1 fifarel:teammate ?player2
}
WHERE {
{
    ?player1 a fifaplg:Player .
    ?player2 a fifaplg:Player .
    FILTER(?player1!=?player2)
}
?player1 fifaplp:team ?team1 .
?player2 fifaplp:team ?team1 .
}
"""
"""
```

Através desta operação inferimos a criação de uma relação *fifarel:teammate* entre quaisquer 2 players diferentes e que pertençam à mesma equipa. Esta inferência abrange jogadores suficientes para estabelecer mais de 400 mil relações deste tipo

- **Team Gender, League Gender e Player League**

```
inference_gender_and_leagues = """
PREFIX fifaplg: <http://fifa24/player/guid/>
PREFIX fifalp: <http://fifa24/league/pred/>
PREFIX fifatp: <http://fifa24/team/pred/>

INSERT {
    ?league fifalp:gender ?gender .
    ?player fifalp:league ?league .
    ?team fifatp:gender ?gender .
}
WHERE {
    ?player a fifaplg:Player .
    ?player fifalp:gender ?gender .
    ?player fifalp:team ?team .
    ?team fifatp:league ?league .
} """
```

Aproveitando que as inferências que queremos realizar acabam por conjugar as mesmas entidades várias vezes aproveitamos essa característica para fazer todas as 3 inferências na mesma query reutilizando a filtragem que é feita para cada uma. Assim criamos novas relações de género para equipa e liga que nos serão úteis posteriormente nas queries à wikidata e criamos uma associação entre um jogador e a sua liga que trará utilidade em termos de UX, uma vez que poderemos aceder diretamente à lida de um jogador sem ter de adivinhar a liga em que joga a equipa deste.

4. Novas operações sobre os dados - SPARQL

Como referido no primeiro projeto, as operações sobre a base de dados foram feitas através de SPARQL. Na pasta *myFifa24/app/api* estão localizados os ficheiros python onde todas as queries são construídas e executadas pelas bibliotecas s4api e SPARQLWrapper.

As várias queries estão divididas em diferentes ficheiros, cada um focado em entidades distintas.

- game.py - queries para o jogo
- leagues.py - queries de ligas
- players.py - queries de jogadores e filtros de jogadores
- squads.py - queries de squads
- teams.py - queries de equipas
- wikidata.py - queries realizadas à wikidata

Tendo já praticamente maximizado a exploração das capacidades da linguagem SPARQL para os dados do nosso sistema, as maiores alterações prenderam-se com a otimização de queries já existentes partindo das novas classes inferidas. Por exemplo:

```
def get_players_by_prop(start: int = 0, limit: int = 30, props: dict = None) -> list[dict]:  
  
    order = "DESC(?ovr) ?name"  
    name = ""  
    gender = "?playerid a fifaplg:Player ."  
    nationality = ""  
    ...  
    case "gender":  
        gender = f'?playerid a fifagg:{prop["value"]} .'  
    case "position":  
    ...  
  
        SELECT ?playerid ?name ?flag ?nationality ?league ?leagueName  
?leagueImg ?teamid ?team ?logo ?position ?ovr ?gender ?image ?skills  
?weakfoot ?attwr ?defwr (CONCAT("{{", GROUP_CONCAT(?stat; separator=",  
"), "}}") AS ?stats)  
        WHERE {{  
            {{  
                {gender}  
                ?playerid fifaplp:gender ?genderid .  
                ?genderid fifagp:label ?gender .  
            ...
```

Nesta query que pretende servir de filtro da página inicial com todos os players alteramos a filtragem de género, deixando de ser uma propriedade e passando a ser a própria classe. Isto irá acelerar em muito a obtenção dos resultados uma vez que a dicotomia masculino/feminino é a que consegue cortar mais facilmente opções de resultado e uma vez que é mais rápido e eficiente procurar por instâncias da classe à cabeça do que procurar por todos os predicados de todas as classes.

```
def get_squad_pontuation(guid: str) -> dict:  
    query = f"""  
PREFIX fifarel: <http://fifa24/relationship/>  
PREFIX fifaspp: <http://fifa24/squad_player/pred/>  
PREFIX fifasqp: <http://fifa24/squad/pred/>  
PREFIX fifasqg: <http://fifa24/squad/guid/>
```

```

SELECT (COUNT(?player1) AS ?points)
WHERE {{

    fifasqg:{guid} fifasqp:player ?squadPlayer1 .
    fifasqg:{guid} fifasqp:player ?squadPlayer2 .
    FILTER(?squadPlayer1 != ?squadPlayer2)
    ?squadPlayer1 fifaspp:player ?player1 .
    ?squadPlayer2 fifaspp:player ?player2 .
    ?player1 fifarel:teammate ?player2 .

}
"""

result = select(query)

return int(result[0]["points"]) if result else 0

```

Em termos de novas queries aos dados existentes na base de dados foi criada a query anterior que aproveita a nova relação inferida *fifarel:teammate* para efetuar um cálculo do número de relações desse tipo que existem numa *Squad* e que irá servir para pontuar as squads criadas pelo utilizador.

As queries efetuadas à Wikidata serão apresentadas no ponto seguinte dedicado aos dados externos que foram incluídos no sistema.

5. Integração de dados externos

Para o nosso sistema definimos que seria relevante para complementar os dados já possuímos com novos dados como o treinador de uma equipa, o seu estádio e a sua lotação, e o percurso de ex-equipas que cada jogador fez. Todas estas informações beneficiam mais se a elas estiver associada uma imagem, uma foto ou um logótipo nestes casos. Tendo à nossa disposição duas fontes de dados principais, a Wikidata e a DBpedia, optámos por incluir exclusivamente dados da Wikidata uma vez que esta é muito mais rica em imagens, apresenta informações mais relevantes em cada entidade e possui um número menor de entidades “duplicadas” relativamente à Wikidata. Uma vez que não temos associado o id da wikidata de cada jogador nem de cada equipa na nossa base de dados realizamos uma espécie “pesquisa” pelo nome guardado na base de dados. Isto levanta problemas como a existência de clubes com equipa B, equipas femininas, com estádios e treinadores diferentes ou até mesmo equipas que têm outro nome na wikidata (caso raro). Para minimizar estes problemas utilizamos as inferências de género já enunciadas para utilizar diferentes predicados nas pesquisas, conseguindo fazer uma separação entre equipas masculinas e equipas femininas. As queries da wikidata relativamente aos estádios e aos treinadores são as seguintes:

```

wd = "Q476028"

if gender:

```

```

wd = "Q28140340"

query=f"""
SELECT DISTINCT ?item
WHERE {{ 
    ?item wdt:P31 wd:{wd} .
    ?item rdfs:label ?name .
    FILTER(CONTAINS(?name, "{team_name}"))
}}
ORDER BY ?name
LIMIT 1
"""

```

```

query_2 = f"""
SELECT DISTINCT ?stadium ?stad ?image ?coach ?coachName
?coachImage ?capacity
WHERE {{ 
    OPTIONAL{{
        <{team_q}> wdt:P115 ?stadium .
        ?stadium rdfs:label ?stad .
        FILTER(LANG(?stad) = "en")
    }}
    OPTIONAL{{
        ?stadium wdt:P18 ?image .
    }}
    OPTIONAL{{
        ?stadium wdt:P1083 ?capacity .
    }}
    OPTIONAL{{
        <{team_q}> p:P286 ?card .
        ?card ps:P286 ?coach .
        OPTIONAL{{
            ?card pq:P580 ?start_date .
        }}
        OPTIONAL{{
            ?card pq:P582 ?end_date .
        }}
        ?coach rdfs:label ?coachName .
        FILTER(LANG(?coachName) = "en")
    }}
    OPTIONAL{{
        ?coach wdt:P18 ?coachImage .
    }}
}}

```

```

    } }

ORDER BY DESC(?start_date) DESC(?end_date) ?coachName
LIMIT 1
"""

```

A estratégia de efetuar 2 queries está diretamente relacionada com a quantidade de dados existentes na Wikidata. Apesar de normalmente só existir um clube com o seu nome uma query normalmente verifica todas as instâncias da classe clube à procura de mais, demorando muito tempo nessa pesquisa. Assim limitamos ao primeiro resultado obtido e assumimo-lo como correto para fazer a query que queremos, desta vez utilizando já o id do clube na wikidata para obter as restantes informações. Na segunda query assumimos que todos os dados são opcionais uma vez que não é garantido que todas as instâncias de equipa tenham aqueles predicados e por isso a utilização constante da instrução OPTIONAL. Também limitamos as labels apenas àquelas que estão em inglês porque cada treinador e estádio pode ter diferentes nomes mediante o idioma em que está escrito e o inglês, enquanto linguagem universal, garante praticamente a existência de uma label nessa língua.

Como o predicado fornecido pela wikidata não garante que retorna o treinador mais atual fazemos nós essa ordenação vendo primeiro qual dos treinadores começou mais tarde o seu trabalho no clube e, para o caso da data de começo não estar disponível, ordenamos em seguida pela data de saída e pelo nome do treinador. Podemos ainda verificar que na primeira query o predicado da equipa que pesquisamos varia mediante o género inferido.

Relativamente às equipas onde um jogador já jogou desenvolvemos as seguintes queries:

```

query = f"""
SELECT DISTINCT ?item
WHERE {
    ?item wdt:P106 wd:Q937857 .
    ?item wdt:P1559 ?o .
    ?item rdfs:label ?name .
    FILTER(LANG(?name) = "en")
    FILTER(CONTAINS(?name, "{player_name}"))
}
LIMIT 1
"""

query_2 = f"""
SELECT DISTINCT ?team ?teamName ?teamImg
WHERE{
    <{player_q}> p:P54 ?card .
    ?card ps:P54 ?team .
    OPTIONAL {
        ?card pq:P580 ?start_date .
    }
}

```

```

OPTIONAL{ 
    ?card pq:P582 ?end_date .
} }
?team rdfs:label ?teamName .
FILTER(LANG(?teamName) = "en")
OPTIONAL{ 
    ?team wdt:P154 ?teamImg .
} }
} }
ORDER BY ?start_date ?end_date
"""

```

Mantendo a estratégia de duas queries para acelerar a obtenção de resultados, pesquisamos numa primeira fase pelo id do jogador que possui um determinado nome. Infelizmente não existe distinção na wikidata entre jogador masculino e feminino. Posteriormente utilizamos esse id para procurar os clubes em que o jogador já jogou, ordenados por ordem crescente de começo de funções, mais uma vez em regime opcional uma vez que cada clube pode não ter todos os dados que gostaríamos de ter.

Apesar dos resultados devolvidos pela query corresponderem àquilo que pedimos são retornadas equipas que na prática são a mesma, por exemplo o jogador Lionel Messi aparece como tendo jogado seguidamente nos clubes FC Barcelona Juvenil A, FC Barcelona C, FC Barcelona Atlètic e FC Barcelona. Para além disto temos registos como as seleções nacionais em que este jogou, que mais uma vez apresenta bastantes escalões não profissionais e equipas olímpicas.

Para filtrar estas equipas, que não são do nosso interesse, realizamos uma filtragem já em python:

```

for result in results:
    if "national" not in result["teamName"].value.lower() and
"olympic" not in result["teamName"].value.lower():
        team_names.append(result["teamName"].value)
        team_imgs.append(result["teamImg"].value if "teamImg"
in result else None)
        team_urls.append(result["team"].value)

last_team = "00000000"
for team in reversed(list(team_names)):
    if last_team in team:
        team_imgs.pop(team_names.index(team))
        team_urls.pop(team_names.index(team))
        team_names.remove(team)
    else:
        last_team = team

```

Desta maneira iremos considerar apenas as equipas que não contenham os termos “national”, para seleções nacionais, e “olympic”, para seleções olímpicas, como sendo clubes a que os jogadores realmente pertenceram. Por fim verificamos se os clubes mais antigos da carreira dos jogadores pertencem ao mesmo, verificando se, numa lista invertida, existem clubes cujo nome começa com o mesmo nome do clube anterior, e em caso afirmativo removemos esses clubes da lista.

6. Publicação de Dados Semânticos

Para publicação dos dados semânticos das nossas páginas optámos por usar apenas RDFa em detrimento da utilização de micro-formatos pois vimos mais vantagens em utilizar um método que se adaptasse mais facilmente e que pudesse abranger mais estruturas de dados como as que foram colocadas nas nossas páginas web. Todas as páginas podem ser encontradas na pasta `/myFIFA24/app/templates`. Foi colocada publicação semântica de dados nas páginas:

- `/player/<guid>/` (template player.html)

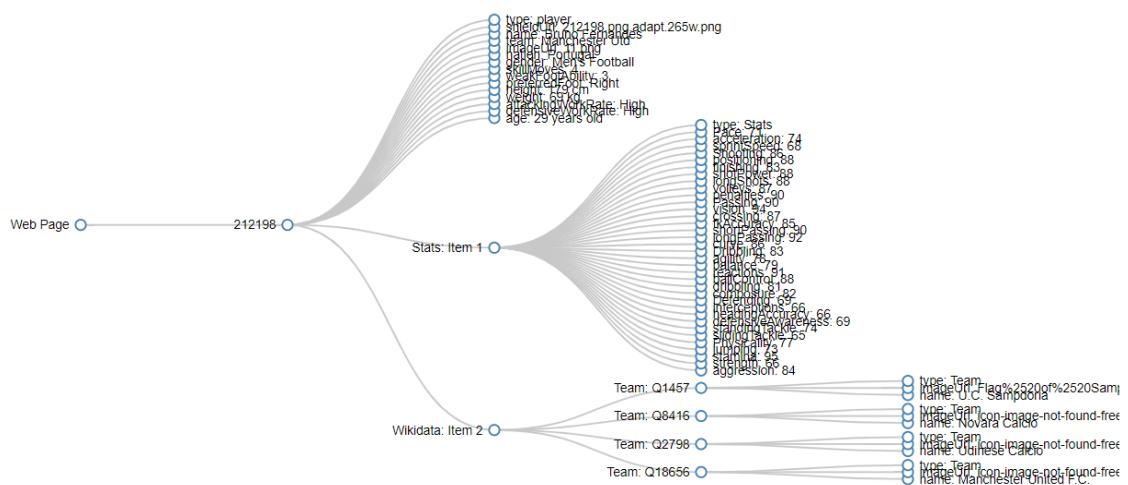


Figura 2 - RDFa da página de Player (alguns nós minimizados para facilitar leitura)

- `/leagues/` (template leagues.html)

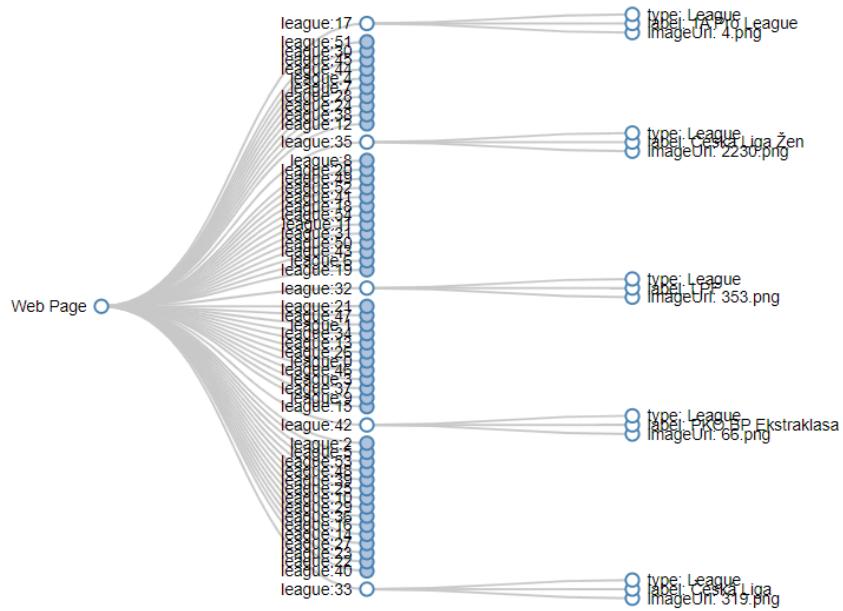


Figura 3 - RDFa da página de leagues (alguns nós minimizados para facilitar leitura)

- **/league/<guid>/** (template league.html)

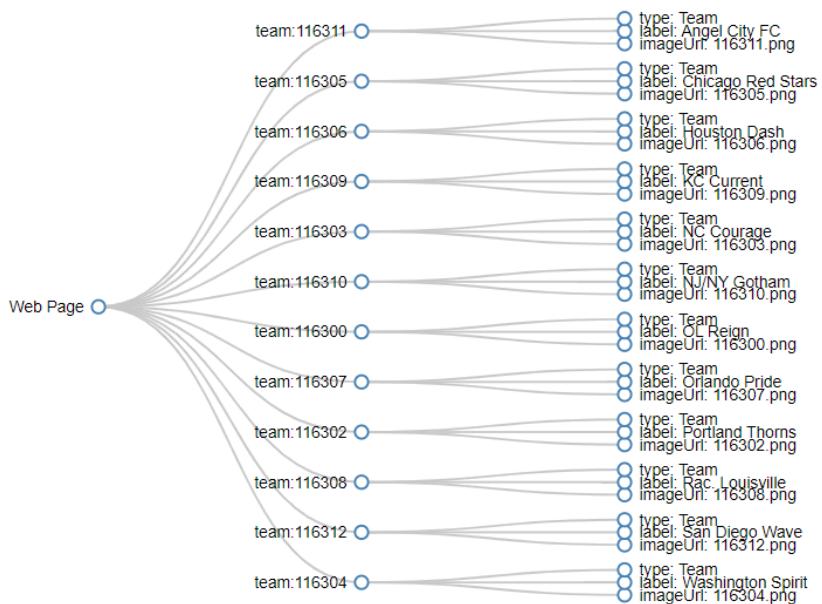


Figura 4 - RDFa da página de league

- **/team/<guid>/** (template team.html)

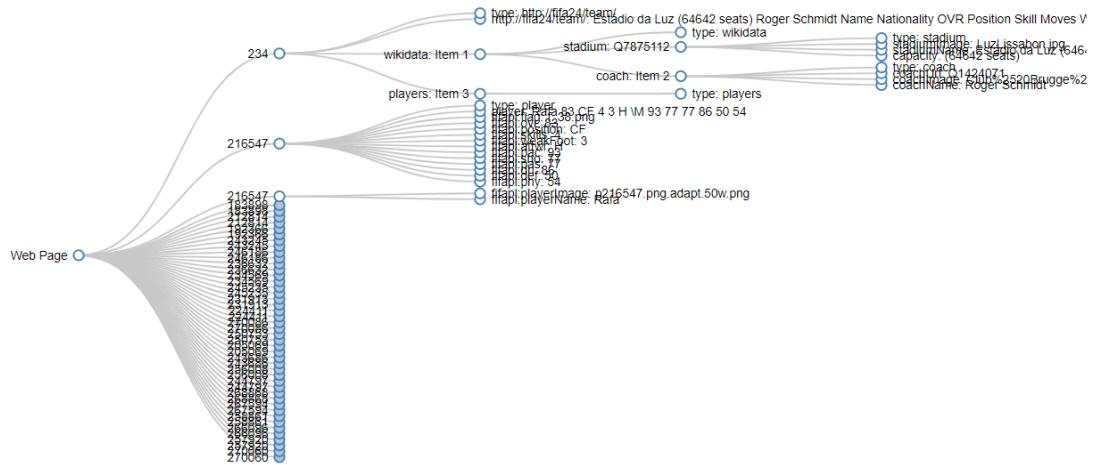


Figura 5 - RDFa da página de team (alguns nós minimizados para facilitar leitura)

- **/players** (template players.html)

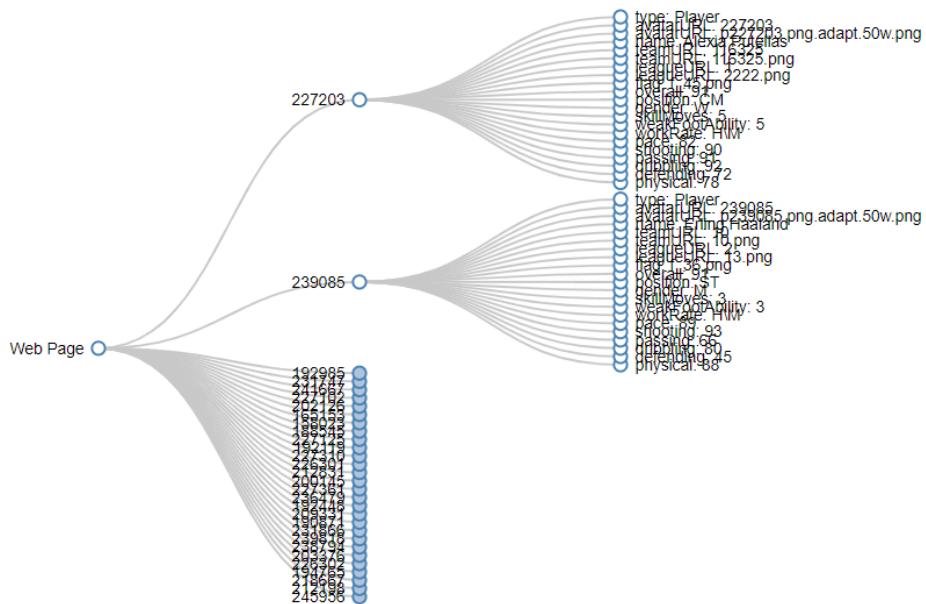


Figura 6 - RDFa da página de players (alguns nós minimizados para facilitar leitura)

7. Funcionalidades da Aplicação

Em termos de funcionalidades a nossa aplicação conta com todas as apresentadas no relatório referente ao projeto anterior e outras novas que são as seguintes:

1. Listar jogadores e ordená-los/filtrá-los por diversas opções

MyFifa24

Players Leagues Game My Squads

Nationality: Team: Gender: Position:

Select Nationality Select Team Select Gender Select Position

	Search by name	Team	League	Nationality	OVR	Position	Gender	Skill Moves	Week Foot Ability	Work Rate	PAC	SHO	PAS	DRI	DEF	PHY
	Alexia Putellas				91	CM	W	5★	5★	H\M	82	90	91	92	72	78
	Erling Haaland				91	ST	M	3★	3★	H\M	89	93	66	80	45	88
	Kevin De Bruyne				91	CM	M	4★	5★	H\M	72	88	94	87	65	78
	Kylian Mbappé				91	ST	M	5★	4★	H\L	97	90	80	92	36	78
	Aitana Bonmatí				90	CM	W	4★	5★	H\H	81	84	83	91	75	73
	Caroline Graham Hansen				90	RW	W	5★	4★	H\M	89	86	88	90	47	75
	Harry Kane				90	ST	M	3★	5★	H\H	69	93	84	83	49	83
	Karim Benzema				90	CF	M	4★	4★	M\M	79	88	83	87	39	78

Esta página já existente no projeto anterior agora acrescenta a liga e o logótipo da mesma que foi inferida daquele jogador e que permite criar um atalho de acesso rápido a essa liga. Anteriormente nada ligava um jogador à sua liga e a ligação entre clube e liga era unidirecional. Desta maneira agora podemos aceder aos clubes de uma liga sem ter de saber a que liga pertence o clube do jogador. Para além disso a pesquisa nesta página tornou-se mais eficiente quando utilizado o filtro de género uma vez que a ontologia permitiu uma estruturação dos dados mais eficiente nessa que reduz em muito os resultados das pesquisas e mais rapidamente. O sistema de cache já implementado para esta página agora abrange todas as queries feitas no sistema e tem um sistema de limpeza que ativa uma thread de 60 em 60 segundos e que verifica se os elementos presentes na cache não são usados há mais de 10 minutos.

Erling Haaland

	PAC	SHO	PAS	DRI	DEF	PHY
Name	Erling Haaland					
Team	Manchester City					
Nation	Norway					
Gender	Men's Football					
Skills	Weak Foot	3★				
Foot	Left	3★				
Height	195 cm					
Weight	94 kg					
Att. WR	High					
Def. WR	Medium					

Pace 89 **Shooting** 93 **Passing** 66

Dribbling 80 **Defending** 45 **Physicality** 88

Acceleration: 82 Positioning: 96 Vision: 74
 Sprint Speed: 94 Finishing: 96 Crossing: 47
 Shot Power: 94 FK Accuracy: 62 Long Shots: 86
 Long Shots: 86 Short Passing: 77 Volleys: 90
 Volleys: 90 Long Passing: 53 Penalties: 84
 Penalties: 84 Curve: 77

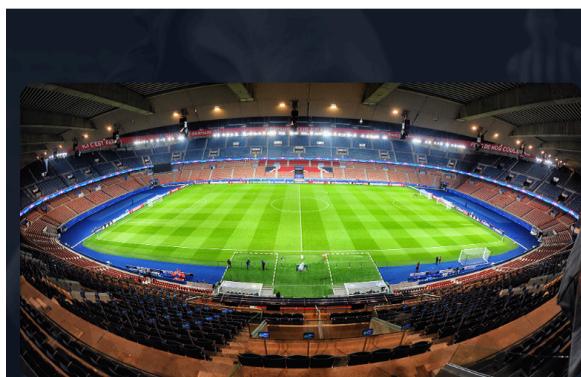
Agility: 76 Interceptions: 43 Jumping: 93
 Balance: 72 Heading Accuracy: 83 Stamina: 76
 Reactions: 94 Def. Awareness: 38 Strength: 93
 Ball Control: 82 Standing Tackle: 47 Aggression: 87
 Dribbling: 79 Sliding Tackle: 29 Sliding Tackle: 29
 Composure: 87

Bryne FK Molde FK FC Red Bull Salzburg Borussia Dortmund Manchester City F.C.

Na página de cada jogador, já existente do projeto anterior também, agora é possível visualizar os dados recuperados da Wikidata, nomeadamente as equipas onde o jogador jogou durante a sua carreira por ordem de transferência. Todos os dados relativos a este assunto são recuperados da wikidata pelo que é normal que existam clubes sem um logótipo associado ou até mesmo jogadores menos conhecidos que nem sequer têm disponível essa informação na wikidata. É possível aceder à página da wikidata de cada clube carregando na imagem ou no nome do clube.

2. Novos Detalhes de Equipa

Quando seleccionamos a equipa que queremos consultar agora, para além da listagem dos jogadores que pertencem a essa equipa, podemos visualizar também o estádio em que a equipa joga, a capacidade em número de lugares e o treinador da mesma. Os conflitos entre estádios e treinadores de equipas femininas e masculinas são resolvidos alterando o predicado na query à wikidata, como já referido. No entanto é possível que um clube não tenha informações registadas/atualizadas na wikidata sobre qual o seu estádio ou qual o seu treinador atual, ou sequer que estes tenham uma foto.



Parc des Princes (47930 seats)



Luis Enrique

Name	Nationality	OVR	Position	Skill Moves	Week Foot Ability	Work Rate	PAC	SHO	PAS	DRI	DEF	PHY
Kylian Mbappé		91	ST	5★	4★	H \L	97	90	80	92	36	78
Gianluigi Donnarumma		87	GK	1★	3★	M \M	90	82	79	89	52	83

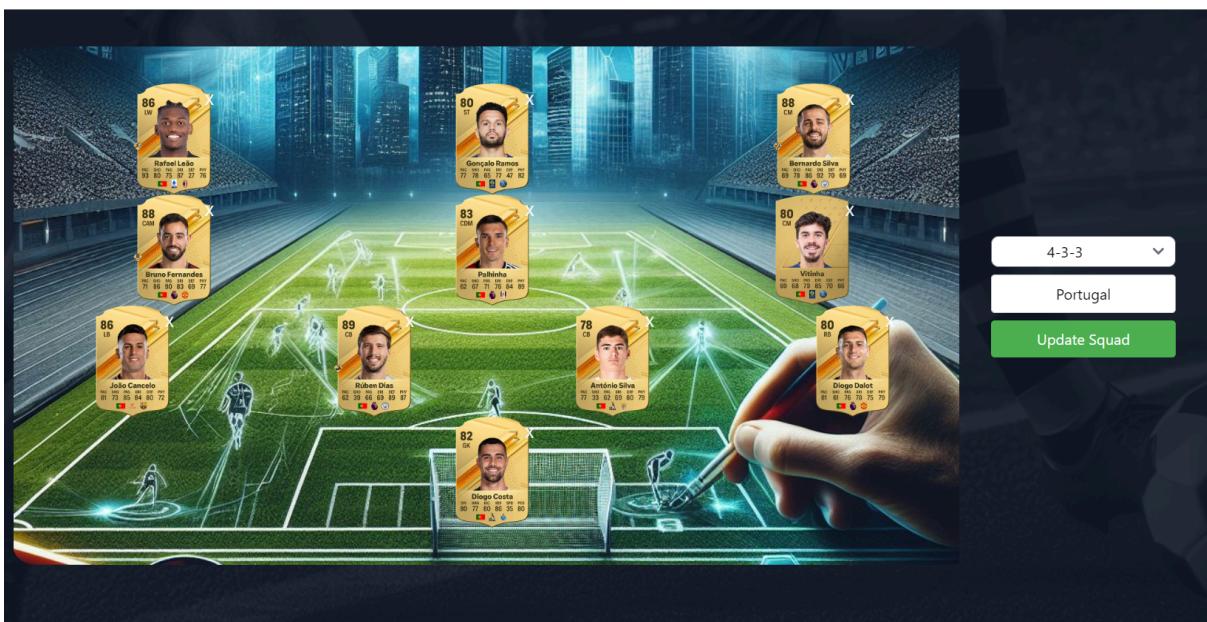
3. Aceder e editar squads criados pelo utilizador

Na secção “My Squads” está disponível uma listagem das squads associada ao nosso utilizador e uma opção para criar uma equipa com os jogadores favoritos do usuário. É também possível eliminar squads previamente criadas através do botão vermelho à frente de cada squad. Para além disto agora cada squad criado pelo utilizador apresenta uma pontuação de equipa que consiste na contagem de relações *fifarel:teammate* entre todos os jogadores daquele plantel.



New Squad	Squad Name	Formation	Pontuation
	Portugal	4-3-3	6

Planeada anteriormente mas não concretizada com sucesso agora é possível alterar qualquer squad criado, quanto ao seu nome, formação e jogadores. Para aceder a esta página e interagir com os jogadores e outros campos basta carregar no nome da squad previamente criada e listada.



8. Executar Aplicação

Pré-requisitos:

1. Instale os requirements do ficheiro *requirements.txt* na pasta *dataset/* e correr o ficheiro *fifa.py* (pode demorar alguns minutos na primeira execução)
2. Instale os requirements do ficheiro *requirements.txt* na pasta *myFIFA24/*

Preparar o GraphDB:

1. Aceder ao ficheiro *GraphDB Desktop.cfg* na pasta *C:\Users\<User>\AppData\Local\GraphDB Desktop\app* e adicionar na última linha a opção “java-options=-Xmx8g”.
2. Inicialize o GraphDB. Caso não inicialize alterar o valor adicionado no passo anterior para um valor superior a 8.
3. Crie um novo repositório chamado *fifa24* com Ruleset OWL2-RL
4. Importe o ficheiro *fifa.n3* da pasta *dataset/* (pode demorar alguns minutos)
5. Importe o ficheiro *ontology.n3* da pasta *dataset/* (ontologia)
6. Executar o programa *dataset/inferences.py*

Executar a aplicação:

1. No diretório *myFIFA24/*, execute o seguinte comando:

```
python manage.py migrate
```

2. Execute o seguinte comando para iniciar o servidor:

```
python manage.py runserver
```

3. Criar uma conta no sistema, fazer login e está pronto para explorar a aplicação.

9. Trabalho Futuro

Com o dataset escolhido ainda existem algumas relações de dados que se podem explorar e ponderar em trabalho futuro.

Adicionar mais formações para o criador de squads seria uma boa adição para o website. Aproveitar as relações entre jogadores como a nova relação de teammates e outras como a nacionalidade para melhorar a pontuação de cada Squad também é uma funcionalidade relevante a implementar tendo em conta o contexto do projeto.

10. Conclusão

Este trabalho proporcionou uma experiência interessante na exploração e consolidação de conhecimentos adquiridos durante Unidade Curricular, nomeadamente em relação a RDFS, OWL, SPIN, Wikidata/fontes externas e RDFa.

O trabalho permitiu ao grupo explorar dentro do possível cada uma das tecnologias utilizadas, obtendo assim um conhecimento prático e teórico maior de cada uma delas. A variedade de dados no dataset foi algo que se revelou desafiante no sentido de tentar inferir novos partindo apenas da ontologia dados ainda não presentes e que tivessem alguma relevância para o objetivo do sistema. Por outro lado a diversidade de informação existente e a maneira como esta foi colocada no sistema permitiu explorar com bastante sucesso as restrições de classes que são possíveis de implementar com OWL, permitiu inferir uma quantidade razoável de novas e importantes relações com SPIN que ajudaram bastante na eficiência das queries feitas à Wikidata e que abriram uma porta bastante interessante para a exploração da “Química” de uma Squad, característica essa muito relevante no contexto em que o nosso sistema se encontra. A quantidade e variedade de páginas, mais propriamente a disponibilização de informação em cada uma, permitiram também uma desafiante exploração da publicação de dados semânticos com RDFa que não tinha sido possível realizar com os exemplos mais simples utilizados nas aulas.

Referências

- [1] “EA Sports FC 24 player ratings database - Electronic Arts.” EA,
<https://www.ea.com/games/ea-sports-fc/ratings>. Accessed 16 April 2024.
- [2] *Drop API Service*, <https://drop-api.ea.com/api>. Accessed 16 April 2024.
- [3] “FUTBIN.” *FUTBIN: EA FC 24 Ultimate Team Prices, Squad Builder, Draft and Players Database*, <https://www.futbin.com>. Accessed 16 April 2024.