

Ultimate Fighting Championship Stats

Etienne Costa
Bruno Sousa
Universidade do Minho
Escola de Engenharia
Braga

(28 de junho de 2021)

Resumo

O presente trabalho é referente a criação de um web site capaz de fornecer as estatísticas gerais do *UFC*, que é a maior organização de artes marciais mistas e ainda ter a capacidade de prever resultados de diversas lutas, tirando partido de uma base de dados semântica baseada em grafos, na qual será possível efectuar diversas queries em SPARQL.

1 Introduction

O presente relatório é referente ao trabalho prático desenvolvido na Unidade Curricular *Processamento e Representação de Conhecimento*, do 4º ano do Mestrado Integrado em Engenharia Informática da Universidade do Minho .

É pretendido com este trabalho prático desenvolver um sistema de consulta de dados estatísticos da maior organização de artes marciais mistas e ainda ter a capacidade de prever resultados de diversas lutas .

Para a realização deste projecto, serão utilizadas diversas ferramentas, por isso numa primeira fase será feita uma contextualização do problema e o conjunto de requisitos à alcançar . De seguida, é feita uma explicação mais detalhada de todo o processo de desenvolvimento,concretamente sobre a obtenção de dados, conversão de dados, funcionamento da arquitetura de todo o sistema,esmiuçando as respectivas componentes de ambos bem como os seus propósitos.

Por fim, será feita uma representação dos resultados obtidos, e uma conclusão perspectivando o trabalho futuro .

2 Analysis and Specification

2.1 Project Description

O projecto em questão consistiu no desenvolvimento de uma aplicação web para realizar consultas de dados estatísticos da maior organização de artes marciais mistas(*UFC*), tendo ainda a possibilidade de prever o resultado de uma luta entre os lutadores presentes no sistema. Tendo sido selecionado um tema, foi necessário fazer a construção de uma ontologia sobre o tema selecionado de modo a armazenar a mesma no *GraphDB*, que é uma base de dados de semântica baseada em grafos, na qual será possível efectuar diversas queries em *SPARQL*.

2.2 Requirement Specification

O levantamento de requisitos é considerada a etapa mais importante, pois é nela que se priorizam as necessidades dos futuros usuários do software, necessidades essas denominadas como *requisitos*. Os principais requisitos levantados nesta etapa foram os seguintes :

Requisitos de descrição:

- Event :
 - Name
 - Date
- Fight :
 - (Red|Blue)CornerSigStr
 - (Red|Blue)CornerSigStrPercentage
 - (Red|Blue)CornerKD
 - (Red|Blue)CornerTD
 - (Red|Blue)CornerTDPercentage
 - (Red|Blue)CornerResult
 - (Red|Blue)CornerCTRL
 - (Red|Blue)CornerReversal
 - (Red|Blue)CornerTotalStrikes
 - (Red|Blue)CornerSubAtt
 - (Red|Blue)Corner
 - Round
 - Bout
 - Method
 - Time
 - TimeFormat
- Fighter :
 - Name
 - Nickname
 - DateOfBirth
 - Weight
 - Height
 - Reach
 - Stance
 - Belt
 - Wins
 - Losses
 - Draws

- Significant Strikes Absorbed per Minute
- Significant Strikes Landed per Minute
- Average Submissions Attempted per 15 minutes
- Significant Strike Defence
- Significant Striking Accuracy
- Takedown Accuracy
- Average Takedowns Landed per 15 minutes
- Takedown Defense
- Referee:
 - Name
- Location:
 - Country
 - City
 - State

Requisitos de Exploração :

- Listar todos os eventos do sistema
- Listar todas as lutas de um determinado evento
- Listar todas os lutadores do sistema
- Listar todas as lutas de um determinado lutador
- Listar as estatísticas gerais por carreira
- Listar as estatísticas gerais por luta
- Listar as estatísticas gerais por evento
- Listar as estatísticas gerais por localização
- Listar as estatísticas gerais por arbitragem
- Listar os campeões actuais da organização
- Fazer a previsão de lutas entre dois lutadores
- Listar o resultado de todas as previsões efectuadas no sistema
- Listar as reviews feitas no sistema
- Listar as estatísticas gerais do repositório

3 Web Scraping

O conjunto de dados era um requisito primordial para a realização deste projeto, sendo que os mesmos podiam ser obtidos de inúmeras maneiras,e.g, **LinkedIn Open Data Cloud,Kaggle**, etc .

Contudo optou-se por ir no sentido inverso e testar as capacidades de produzir um ficheiro semiestruturado recorrendo ao web scraping . O **web scraping** é uma solução para quem quer ter acesso a dados da web de uma forma automatizada, principalmente quando o site do qual se pretende obter dados não possui uma API.

No contexto do nosso problema tivemos sempre presente a informação relevante a ser extraída,i.e, o conjunto de eventos decorridos no *ufc* desde 1994 até ao ano corrente, bem como a informação de cada uma das lutas e dos seus intervenientes

Portanto seguiu-se o processo típico de web scrapping :

- Identificar os websites
 - <http://ufcstats.com/statistics/events/completed>
 - <http://ufcstats.com/statistics/fighters>
 - <https://www.ufc.com/athletes/all>
- Recolher URLs das páginas identificadas
- Fazer um pedido a estes URLs de modo a obter o HTML da página
- Utilizar localizadores para encontrar os dados em HTML
- Armazenar os dados num ficheiro JSON e CSV

3.1 Scripting

A linguagem de programação escolhida para enfrentar este desafio foi o *Python*, pois a mesma possuí diversas bibliotecas que nos auxiliam na execução de diversas tarefas,e.g., *requests*, *re*, *json*, *os*, *time* e *BeautifulSoup* e além disso acaba por ser uma linguagem extremamente fácil para trabalhar.

Portanto optou-se por fazer 6 scripts, tendo cada um deles o seguinte objectivo :

- events.py - responsável por obter toda a informação dos eventos e as suas respectivas lutas .
- fighters.py - responsável por obter a informação de todos os lutadores .

- face.py - responsável por obter as fotos de perfil existentes de todos os lutadores .
- full-body.py - responsável por obter as fotos de todos os lutadores .
- auxiliar.py - responsável por obter a quantidade de dados de modo a calcular a barra de progresso .
- ufc.py - script responsável por fundir toda informação e gerar um ficheiro semiestruturado .

3.2 JavaScript Object Notaion

A execução do *ufc.py* permite-nos obter um json com o seguinte formato :

```
{
  "Referees": [
    {
      "Name": "Chris Tognoni"
    },
    {
      "Name": "Bill Clancy"
    },
    ". . ."
  ],
  "Fighters": [
    {
      "Name": "Khabib Nurmagomedov",
      "Nickname": "The Eagle",
      "Height": "5' 10",
      "Weight": "155 lbs",
      "Reach": 70.0,
      "Stance": "Orthodox",
      "Wins": 29,
      "Losses": 0,
      "Draws": 0,
      "Belt": "false",
      "DateOfBirth": "Sep 20 1988",
      "SLpM": 4.1,
      "StrAcc": 48.0,
      "SApM": 1.75,
      "StrDef": 65.0,
      "TDAvg": 5.32,
      "TDAcc": 48.0,
      "TDDef": 84.0,
      "SubAvg": 0.8
    },
    ". . ."
  ],
  "Events": [

```

```

{
  "Name": "UFC_Fight_Night_Reyes_vs_Prochazka",
  "Location": {
    "City": "Las Vegas",
    "State": "Nevada",
    "Country": "USA"
  },
  "Date": "May 01 2021",
  "Fights": [
    {
      "Bout": "Light Heavyweight Bout",
      "RedCorner": "Dominick Reyes",
      "BlueCorner": "Jiri Prochazka",
      "RedCornerResult": "L",
      "BlueCornerResult": "W",
      "Method": "KO/TKO",
      "Round": "2",
      "Time": "4:29",
      "TimeFormat": "5 Rnd (5-5-5-5-5)",
      "Referee": "Herb Dean",
      "Stats": {
        "RedCorner": {
          "KD": "0",
          "SigStr": "63 of 108",
          "SigStrPercentage": "58%",
          "TotalStrikes": "68 of 114",
          "TD": "1 of 1",
          "TDPercentage": "100%",
          "SubAtt": "1",
          "Reversal": "0",
          "CTRL": "0:29"
        },
        "BlueCorner": {
          "KD": "1",
          "SigStr": "77 of 136",
          "SigStrPercentage": "56%",
          "TotalStrikes": "78 of 137",
          "TD": "1 of 1",
          "TDPercentage": "100%",
          "SubAtt": "0",
          "Reversal": "0",
          "CTRL": "1:35"
        }
      }
    },
    ". . ."
  ]
},
". . ."
]

```

}

Listing 1: Estrutura do json

3.3 JSON Fields

Alguns termos utilizados nos campos do json acabam por ser estranhos a quem não está familiarizado com o meio das lutas, portanto é feita uma breve tradução dos mesmos :

- SLpM : Significant Strikes Landed Per Minute
- StrAcc : Significant Striking Accuracy
- SApM : Significant Strikes Absorbed Per Minute
- StrDef : Significant Strike Defence
- TDAvg : Average Takedowns Landed Per 15 Minutes
- TDAcc : Takedown Accuracy
- TDDef : Takedown Defense
- SubAvg : Average Submissions Attempted per 15 minutes
- KD : KnockDowns
- SigStr : Significant Strikes
- SigStrPercentage : Significant Strikes Percentage
- TD : Takedowns
- TDPercentage : Takedowns Percentage
- SubAtt : Submissions Attempted
- CTRL : Control Time

3.4 Data Stats

A organização do *ufc* realiza eventos semanalmente, exigindo uma constante actualização da informação, portanto seria possível fazer uma execução semanal do nosso script de modo a obter os dados mais recentes, porém, visto que são feitos inúmeros pedidos para obter a informação dos eventos e lutadores o script leva um tempo considerável até obter toda informação então a versão apresentada neste momento foi obtida no dia 18 de Junho de 2021 tendo no total a seguinte informação :

Events	Fights	Fighters	Referees	Locations
565	6149	3664	207	166

Tabela 1: Overall Stats

Informação essa calculada à custa de um pequeno script :

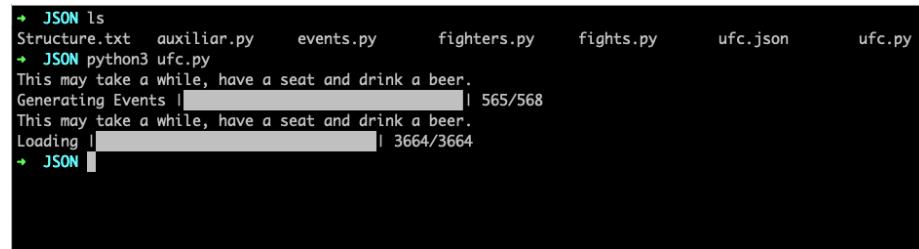
```
#!/usr/bin/python3

import json
with open('../web-scraping/JSON/ufc.json','r') as ufcfile:
    data = json.load(ufcfile)
    referees = data['Referees']
    fighters = data['Fighters']
    events = data['Events']
    fights = 0
    locations = []

for e in events:
    if e['Location'] not in locations:
        locations.append(e['Location'])
    fights+=len(e['Fights'])

print("Referees :: ",len(referees))
print("Fighters :: ",len(fighters))
print("Events :: ",len(events))
print("Fights ::", fights)
print("Locations ::",len(locations))
```

Listing 2: stats.py



```
➜ JSON ls
Structure.txt auxiliar.py events.py fighters.py fights.py ufc.json ufc.py
➜ JSON python3 ufc.py
This may take a while, have a seat and drink a beer.
Generating Events |██████████| 565/568
This may take a while, have a seat and drink a beer.
Loading |██████████| 3664/3664
➜ JSON
```

Figura 1: Creating JSON Script

4 Ontology

Uma ontologia é um modelo de dados que representa um conjunto de conceitos dentro de um domínio e os relacionamentos entre estes, sendo que a mesma pode ser vista como uma especificação formal de conhecimento,i.e, capaz de ser compreendida por humanos e máquinas .

Ontologias geralmente descrevem :

- Indivíduos : Objectos básicos
- Classes : Conjuntos, coleções ou tipos de objectos
- Atributos : Propriedade,características ou parâmetros que os objectos podem ter e compartilhar
- Relacionamentos : As formas como os objectos se podem relacionar com outros objectos

4.1 Specifying Ontologies

Para especificar a nossa ontologia foram seguidas as seguintes metodologias:

1. Especificação do domínio :
 - Para que é que vamos usá-la?
 - A que perguntas deve dar resposta?
 - Quem vai usá-la e mantê-la ?
2. Enumeração dos termos mais importantes do domínio
3. Definição das classes
4. Definição dos atributos de cada classe
5. Definição das relações entre indivíduos
6. Definição dos indivíduos

4.1.1 Especificação do domínio

A nossa ontologia foi definida com o propósito de proporcionar a informação pertinente do conjunto de atletas que pertencem ao *ufc*,i.e, providenciar um conjunto de dados estatísticos sobre todas as classes, podendo construir uma api capaz de consumir toda a informação da mesma e ainda conseguir fazer uma previsão de um possível resultado entre dois atletas com base nos seus atributos .

A manutenção da mesma será providenciada pelos seus criadores, tendo o cuidado de fazer um possível refactoring das queries em SPARQL de modo a não se perder o desempenho durante futuras consultas .

4.1.2 Termos do domínio

Tendo feito a especificação dos termos do nosso domínio, é possível identificar o conjunto de entidades, propriedades e relacionamentos entre os indivíduos .

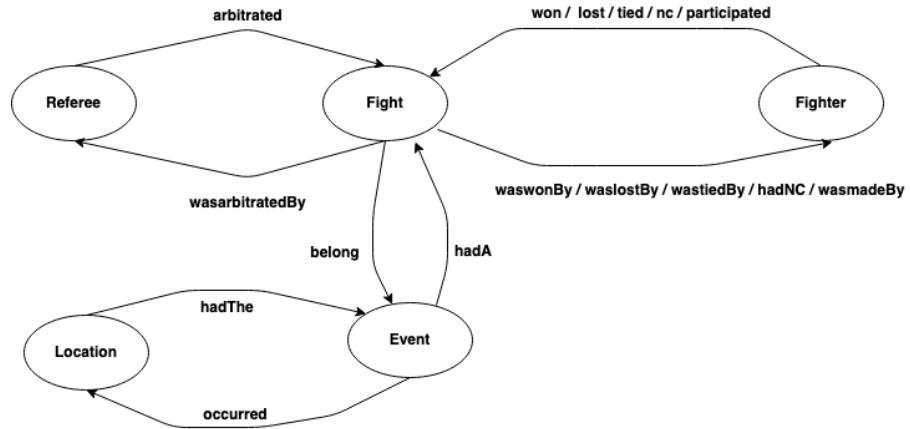


Figura 2: Termos do domínio

Classes :

- Event
 - Fight
 - Fighter
 - Referee
 - Location
-

Listing 3: Classes

```
Data Properties :
    - Event :
        - Date
        - Name
    - Fight :
        - (Red|Blue)CornerSigStr
        - (Red|Blue)CornerSigStrPercentage
        - (Red|Blue)CornerKD
        - (Red|Blue)CornerTD
        - (Red|Blue)CornerTDPercantage
        - (Red|Blue)CornerResult
        - (Red|Blue)CornerCTRL
        - (Red|Blue)CornerReversal
        - (Red|Blue)CornerTotalStrikes
        - (Red|Blue)CornerSubAtt
        - (Red|Blue)Corner
        - Round
        - Bout
        - Method
        - Time
        - TimeFormat
    - Fighter :
        - Name
        - Nickname
        - DateOfBirth
        - Weight
        - Height
        - Reach
        - Stance
        - Belt
        - Wins
        - Losses
        - Draws
        - SAPM
        - SLPM
        - SubAvg
        - StrDef
        - StrAcc
        - TDAcc
        - TDAvg
        - TDDef
    - Referee :
        - Name
    - Location :
        - Country
        - City
        - State
```

Listing 4: Data Properties

Object Properties :

Pode ser interpretado como, um árbitro arbitrou uma luta, sendo o wasarbitratedBy a relação inversa .

- arbitrated : Referee --*arbitrated*--> *Fight*
- wasarbitratedBy : *Fight* --*wasarbitratedBy*--> Referee

Pode ser interpretado como, um evento teve uma luta, sendo o belong a relação inversa .

- hadA : Event --*hadA*--> *Fight*
- belong : *Fight* --*belong*--> Event

Pode ser interpretado como, um fighter teve um 'no contest' numa luta, sendo o hadNC a relação inversa .

- nc : Fighter --*nc*--> *Fight*
- hadNC : *Fight* --*hadNC*--> Fighter

Pode ser interpretada como, um evento ocorreu numa localização, sendo o hadThe a relação inversa .

- occurred : Event --*occurred*--> Location
- hadThe : Location --*hadThe*--> Event

Pode ser interpretada como, um lutador perdeu uma luta, sendo o waslostBy a relação inversa .

- lost : Fighter --*lost*--> *Fight*
- waslostBy : *Fight* --*waslostBy*--> Fighter

Pode ser interpretada como, um lutador participou numa determinada luta, sendo o wasmadeBy a relação inversa .

- participated : Fighter --*participated*--> *Fight*
- wasmadeBy : *Fight* --*wasmadeBy*--> Fighter

Pode ser interpretada como, um lutador empatou uma luta, sendo o wastiedBy a relação inversa .

- tied : Fighter --*tied*--> *Fight*
- wastiedBy : *Fight* --*wastiedBy*--> Fighter

Pode ser interpretada como, um lutador ganhou uma luta, sendo o waswonBy a relação inversa .

```
- won :     Fighter --won--> Fight
- waswonBy : Fight --waswonBy--> Fighter
```

Listing 5: Object Properties

4.2 Turtle

O Resource Description Framework (RDF) é uma linguagem de propósito geral para representar informações na web.

Nesta secção é feita a conversão do json para um sintaxe textual de RDF chamada Turtle, que permite que um grafo RDF seja completamente escrito em um formato de texto compacto e natural, com abreviações para padrões de uso e tipos de dados comuns .

4.2.1 JSON To Turtle

Para fazer a conversão fez-se a modelação do modelo base usando o Protégé , e de seguida implementou-se um script em python para fazer a geração do turtle. Durante o processo de geração calculou-se a relação entre as classes num único sentido de modo a fazer a inferência das restantes relações no próprio Protégé .

4.2.2 Extreme Cases

Após a geração do ficheiro turtle, verificaram-se duas situações que comprometiam a utilização do mesmo. A primeira situação estava relacionada com o identificador dos lutadores, pois haviam 5 atletas que possuíam o mesmo nome e visto que nem sempre tínhamos as datas de nascimento, optou-se por fundir as lutas num único lutador e ficar com as data properties que pertencem ao segundo lutador encontrado .

```
Allow us to verify if the file have repeated names .
```

```
$ sort names.txt | uniq -d > repeated.txt
$ cat repeated.txt
```

```
Bruno Silva
Joey Gomez
Michael McDonald
Mike Davis
Tony Johnson
```

Listing 6: Repeated Names

A segunda situação está relacionada com o facto de no início da organização, haviam atletas que no mesmo evento podiam lutar duas vezes e estas lutas podiam assumir qualquer resultado, produzindo assim uma inconsistência no nosso turtle pois num determinado evento haviam lutas que apresentavam vários resultados.

UFC - Ultimate Japan									
		DATE: December 21, 1997 LOCATION: Yokohama, Kanagawa, Japan							
W/L	FIGHTER	KD	STR	TD	SUB	WEIGHT CLASS	METHOD	ROUND	TIME
WIN	Randy Couture Maurice Smith	0 0	19 9	3 0	1 0	Heavyweight KO	M-DEC	3	3:00
WIN	Kazushi Sakuraba Marcus Silveira	0 0	1 1	1 1	2 1	Heavyweight KO	SUB Armbar	1	3:44
WIN	Vitor Belfort Joe Charles	0 0	0 1	2 0	3 0	Heavyweight	SUB Armbar	1	4:03
WIN	Frank Shamrock Kevin Jackson	0 0	1 2	0 1	1 0	Lightweight KO	SUB Armbar	1	0:16
NC	Kazushi Sakuraba Marcus Silveira	0 0	0 4	0 1	0 1	Heavyweight	Overturned	1	1:51
WIN	David Abbott Yoji Anjo	0 0	13 9	3 0	0 1	Heavyweight	U-DEC	2	3:00
WIN	Tra Telligman Brad Kohler	0 0	6 12	0 3	1 0	Heavyweight	SUB Armbar	1	10:05

5 API Server

O servidor que contém a API foi desenvolvido com o *Node.js*, usando a framework de desenvolvimento *Express.js*. Este servidor por sua vez está em constante comunicação o *GraphDB*, visto que o mesmo possuí a nossa ontologia povoada .

Esta API servirá como ponto de ligação dos end-users/aplicações à informação. A mesma não se encontra protegida, porque o propósito é providenciar de forma aberta informações,estatísticas gerais e possíveis previsões de resultados aos principais amantes de artes marciais .

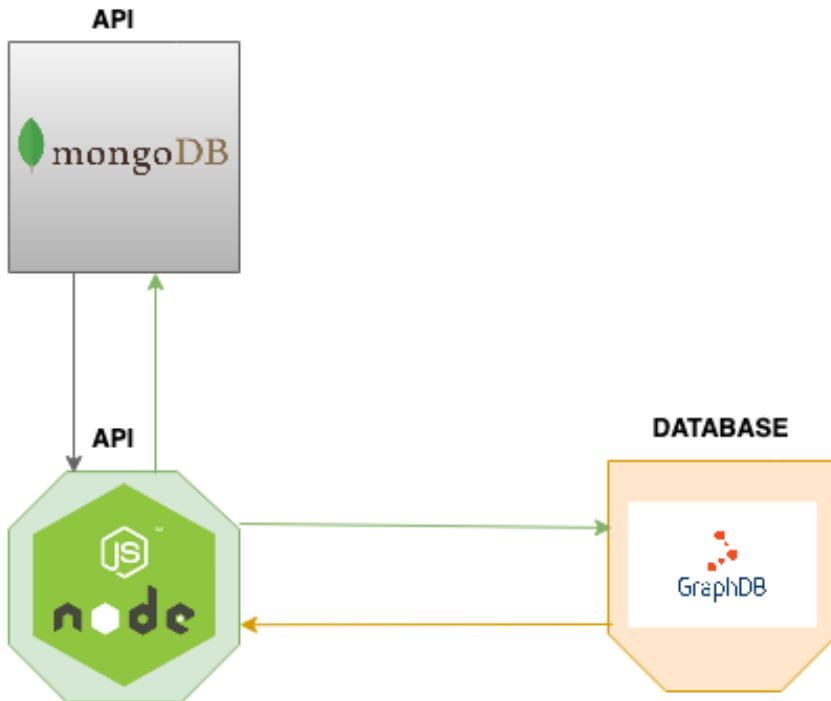


Figura 3: Backend Structure

5.1 Rotas

As rotas encaminham os pedidos dos clientes de forma a realizaram os pedidos a nossa base de dados, sendo que a rota para onde o utilizador é encaminhado varia consoante o URL e o método do pedido.

```
var prefixes = '
PREFIX : <http://www.di.uminho.pt/prc2021/ufc#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```

PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>‘

exports.execQuery = async function (query){
    var getLink = "http://localhost:7200/repositories/UFC?query="
    var encoded = encodeURIComponent(prefixes + query)
    var result = await axios.get(getLink + encoded)
    return result.data
}

```

Listing 7: graphdb.js

De seguida, apresentam-se as rotas definidas na nossa API :

Fighters: `ufc/fighters`:

- método GET
 - / - Devolve a lista de lutadores no ufc com a sua respectiva informação pessoal .
 - /**names** - Devolve a lista de nomes de todos os lutadores no ufc.

Fighter: `ufc/fighter`:

- método GET
 - /**:id** - Devolve a informação pessoal de um determinado lutador.
 - /**fights**/**:id** - Devolve todas as lutas de um determinado lutador.

Stats: `ufc/stats`:

- método GET
 - /**career/ufc-fights** - Devolve o top 10 dos lutadores com mais lutas no ufc.
 - /**career/mma-fights** - Devolve o top 10 das lutas com mais lutas nas artes marciais mistas.
 - /**career/wins** - Devolve o top 10 dos lutadores com mais vitórias no ufc.
 - /**career/title-fight-wins** - Devolve o top 10 dos lutadores com mais vitórias em lutas por um título.
 - /**career/KO-TKO** - Devolve o top 10 dos lutadores com mais vitórias por nocaute.
 - /**career/submission-wins** - Devolve o top 10 dos lutadores com mais vitórias por submissão.

- **/career/finishes** - Devolve o top 10 dos lutadores com mais vitórias por finalização .
- **/career/decision** - Devolve o top 10 dos lutadores com mais vitórias por decisão .
- **/career/strike-accuracy** - Devolve o top 10 dos lutadores com melhor precisão a nível de golpes .
- **/career/strikes landed per min** - Devolve o top 10 dos lutadores com mais golpes lançados por minutos .
- **/career/strike-defense** - Devolve o top 10 dos lutadores com maior precisão de golpes defendidos .
- **/career/takedown-accuracy** - Devolve o top 10 dos lutadores com maior precisão de quedas .
- **/career/takedown-defense** - Devolve o top 10 dos lutadores com maior precisão de quedas defendidas .
- **/career/submission-average** - Devolve o top 10 dos lutadores com a melhor média de submissões .
- **/fight/fastest-finish** - Devolve o top 10 das lutas com menor duração .
- **/fight/fastest-KO-TKO** - Devolve o top 10 das lutas com menor duração terminadas por um nocaute .
- **/fight/fastest-submission** - Devolve o top 10 das lutas com menor duração terminadas por uma submissão .
- **/fight/latest-finish** - Devolve o top 10 das lutas com a finalização mais tardia .
- **/fight/latest-KO-TKO** - Devolve o top 10 das lutas com a finalização mais tardia por nocaute .
- **/fight/latest-submission** - Devolve o top 10 das lutas com a finalização mais tardia por submissão .
- **/fight/knockdowns landed** - Devolve o top 10 das lutas com nocautes .
- **/event/KO-TKO** - Devolve o top 10 dos eventos com maior número de lutas decididas por nocaute .
- **/event/submission-wins** - Devolve o top 10 dos eventos com maior número de lutas decididas por submissão .
- **/event/decision-wins** - Devolve o top 10 dos eventos com maior número de lutas decididas por pontos .
- **/location** - Devolve o top 10 das localizações com mais lutas .
- **/referee** - Devolve o top 10 dos árbitros com mais lutas arbitradas .
- **/titleholders** - Devolve a lista de campeões da organização .

Events: ufc/events:

- método GET
 - / - Devolve a lista de eventos já realizados no ufc .
 - **/details/:id** - Devolve os detalhes de um determinado evento .
 - **/fights/:id** - Devolve as lutas de um determinado evento .

Repository: /ufc/repositorie:

- método GET
 - / - Devolve a informação do repositório .

6 Predictions

A comunidade que acompanha os grandes eventos de luta muita das vezes procura fazer o impossível, que é prever o resultado de uma luta. O problema é que por natureza as lutas são extremamente imprevisíveis, o que torna as mesmas interessantes .

Portanto recorreu-se a um projecto na área de *Machine Learning* desenvolvido por *Charles Pierse* que tirando partido dos dados estáticos e dinâmicos dos lutadores constrói um sistema de previsão bi-direcional que funciona do seguinte modo :

- The first model uses the static stats as the independent variables for fighter 1 and fighter 2 and then predicts the dependent variables, the dynamic fight stats.(**multi target regression**).
- We then pass the static and dynamic fight stats to an overall winner model that predicts whether fighter 1 or fighter 2 is the winner. (**binary classification**).

7 Interface

A interface foi desenvolvida através da framework *Vue.js* e a framework de material *Vuetify.js* baseada no *Material Design da Google*.

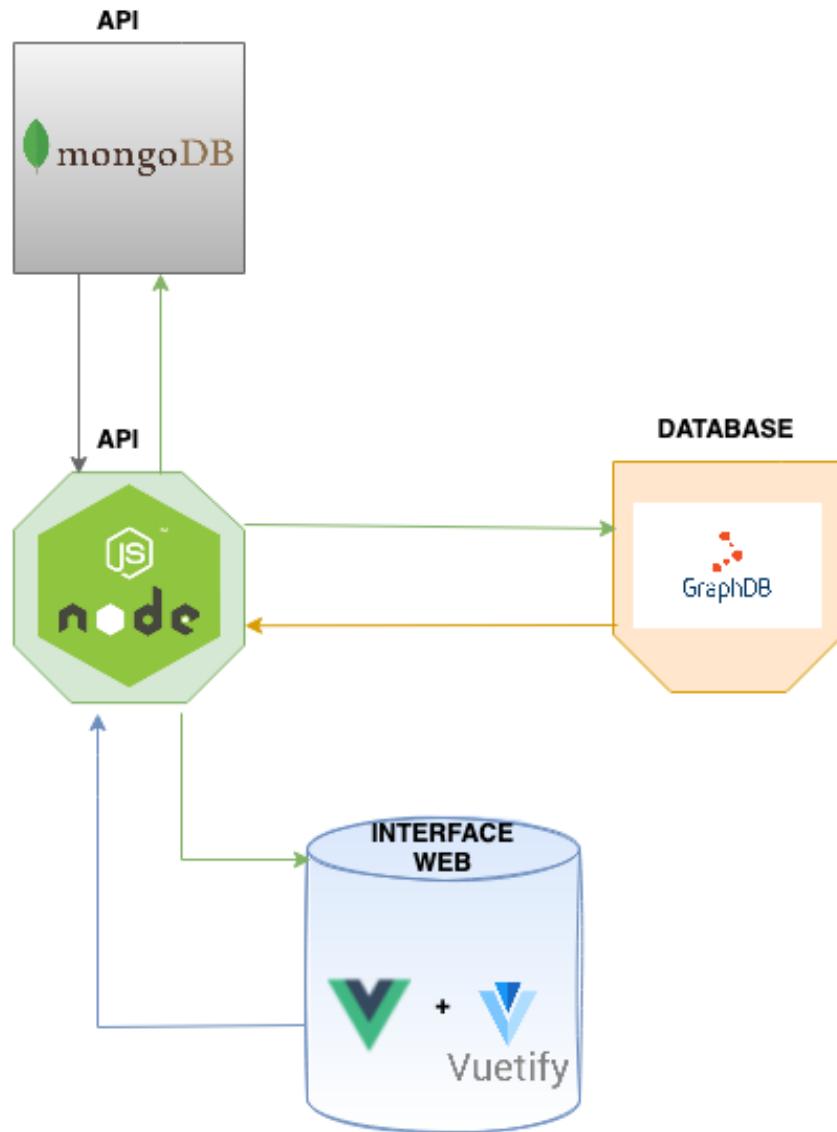


Figura 4: Backend and Frontend Structure

Relativamente ao aspecto do website, procurou-se fazer uma implementação com um aspecto atrativo, recorrendo a inúmeras imagens descarregadas durante

o processo de *scraping*, uma imagem na homepage com um dos maiores atletas da companhia e ícones anexados ao menu da navbar.

De seguida, serão apresentadas as diversas camadas da interface.

7.1 Routes

Nesta secção são apresentadas as rotas definidas :

Rota associada a homepage, sendo feita a renderização da view Home.vue.

```
{  
  path: '/',
  redirect: { name: 'Home' }  
},  
,  
{  
  path: '/ufc',
  name: 'Home',
  component: Home,  
}
```

Listing 8: Homepage route

Rota associada a página dos lutadores, sendo feita a renderização da view Fighters.vue.

```
{  
  path: '/ufc/fighters',
  name: 'Fighters',
  // route level code-splitting
  // this generates a separate chunk (about.[hash].js) for this route
  // which is lazy-loaded when the route is visited.
  component: () => import(/* webpackChunkName: "about" */  
    '../views/Fighters.vue')  
}
```

Listing 9: Fighters route

Rota associada a página dos eventos, sendo feita a renderização da view Events.vue.

```
{  
  path: '/ufc/events',
  name: 'Events',
  // route level code-splitting
  // this generates a separate chunk (about.[hash].js) for this route
  // which is lazy-loaded when the route is visited.
```

```
        component: () => import(/* webpackChunkName: "about" */  
          './views/Events.vue')  
    }  
  
```

Listing 10: Events route

Rota associada a página de um evento, sendo feita a renderização da view Event.vue.

```
{  
  path: '/ufc/event/:id',  
  name: 'Event',  
  // route level code-splitting  
  // this generates a separate chunk (about.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => import(/* webpackChunkName: "about" */  
    './views/Event.vue')  
}  
 
```

Listing 11: Homepage route

Rota associada a página de um lutador, sendo feita a renderização da view Fighter.vue.

```
{  
  path: '/ufc/fighter/:id',  
  name: 'Fighter',  
  // route level code-splitting  
  // this generates a separate chunk (about.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => import(/* webpackChunkName: "about" */  
    './views/Fighter.vue')  
},  
 
```

Listing 12: Homepage route

Rota associada a página de previsões, sendo feita a renderização da view Predictions.vue.

```
{  
  path: '/ufc/predictions/',  
  name: 'Predictions',  
  // route level code-splitting  
  // this generates a separate chunk (about.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => import(/* webpackChunkName: "about" */  
    './views/Predictions.vue')  
}  
 
```

Listing 13: Homepage route

Rota associada a página com o histórico previsões, sendo feita a renderização da view PredictionsHistory.vue.

```
{  
  path: '/ufc/predictions/history',  
  name: 'PredictionsHistory',  
  // route level code-splitting  
  // this generates a separate chunk (about.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => import(/* webpackChunkName: "about" */  
    '../views/PredictionsHistory.vue')  
}
```

Listing 14: Homepage route

Rota associada a página de estatísticas gerais da companhia, sendo feita a renderização da view Stats.vue.

```
{  
  path: '/ufc/stats/',  
  name: 'Stats',  
  // route level code-splitting  
  // this generates a separate chunk (about.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => import(/* webpackChunkName: "about" */  
    '../views/Stats.vue')  
}
```

Listing 15: Homepage route

Rota associada a página de registo de avaliações gerais da aplicação, sendo feita a renderização da view RateUs.vue.

```
{  
  path: '/ufc/rate-us',  
  name: 'Rate',  
  // route level code-splitting  
  // this generates a separate chunk (about.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => import(/* webpackChunkName: "about" */  
    '../views/RateUs.vue')  
}
```

Listing 16: Homepage route

Rota associada a página de avaliações efectuadas da aplicação, sendo feita a renderização da view Reviews.vue.

```
{  
  path: '/ufc/reviews',  
  name: 'Review',  
  // route level code-splitting  
  // this generates a separate chunk (about.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => import(/* webpackChunkName: "about" */  
    '../views/Reviews.vue')  
}
```

Listing 17: Homepage route

Rota associada a página de estatísticas gerais do repositório, sendo feita a renderização da view Ontology.vue.

```
{  
  path: '/ufc/ontology/details',  
  name: 'Ontology',  
  // route level code-splitting  
  // this generates a separate chunk (about.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => import(/* webpackChunkName: "about" */  
    '../views/Ontology.vue')  
}
```

Listing 18: Homepage route

Permite identificar um url inválido na aplicação, renderizando a página com o erro associado (404) .

```
{  
  path: '*',  
  name: 'PageNotFound',  
  // route level code-splitting  
  // this generates a separate chunk (about.[hash].js) for this route  
  // which is lazy-loaded when the route is visited.  
  component: () => import(/* webpackChunkName: "about" */  
    '../views/PageNotFound.vue')  
}
```

Listing 19: Homepage route

7.2 Views

Nesta secção são apresentadas as views definidas :

`Home.vue` :

Está é a vista inicial com qual o utilizador tem a primeira interação, a mesma contém uma navbar e um footer, sendo que estes últimos são fixos,i.e, aparecem em todas as views implementadas. Também foi adicionado um vídeo com alguns highlights das lutas.

Listing 20: Home.vue

`Fighters.vue` :

Nesta vista é possível ver uma tabela com a informação de todos os lutadores, sendo possível fazer pesquisas por qualquer um dos campos presentes no cabeçalho. Alguns dados como height,weight e stance podem ser convertidos para unidades mais convencionais,e.g, lbs par kg .

Listing 21: Fighters.vue

`Fighter.vue` :

Nesta vista é possível ver o cartel actual do lutador,bem como os dados estáticos e dinâmicos do mesmo.
Durante o processo de scraping, foram descarregadas algumas imagens e caso o lutador tenha uma imagem é feita o display da mesma.
Por baixo destas informações conseguimos consultar todas as lutas que o mesmo participou dentro da organização .

Listing 22: Fighter.vue

`Events.vue` :

Nesta vista é possível ver uma tabela com a informação de todos os eventos, sendo possível fazer pesquisas por qualquer um dos campos presentes no cabeçalho.

Listing 23: Events.vue

`Event.vue` :

Nesta vista é possível ver a informação detalhada de um evento em específico,i.e, a data e a localização onde decorreu o evento, bem com a informação de todas as lutas que aconteceram neste evento.

Listing 24: Event.vue

Stats.vue :

Esta vista acaba por ser aquela que processa a maior quantidade de informação face ao número de pedidos que são feitos na nossa API, nela é possível consultar as estatísticas gerais da companhia sendo que as mesmas estão divididas em 5 categorias distintas.

Listing 25: Stats.vue

Predictions.vue :

Nesta view é possível ver a listagem dos nomes dos lutadores, sendo obrigatório selecionar 2 lutadores de modo a conseguir fazer uma previsão do resultado de uma luta .
Durante o processo de scraping foram descarregadas imagens dos lutadores, caso os lutadores selecionados possuam uma imagem é feito o display delas .

Listing 26: Predictions.vue

Ontology.vue :

Nesta view é possível consultar alguma informação do repositório criado no graphDB .

Listing 27: Ontology.vue

PredictionsHistory.vue :

Nesta view é possível consultar o histórico de previsões efectuadas no sistema, visto que as mesmas podem ir mudando com as actualizações efectuadas no sistema .

Listing 28: PredictionsHistory.vue

RateUS.vue :

Nesta view é possível dar um feedback a aplicação, caso sejam fornecidos dados validados pelo formulário disponível .

Listing 29: RateUS.vue

Reviews.vue :

Nesta view é possível consultar todo o feedback dado ao sistema,

sendo que no mesmo aparecem a informação aparece ordenada pela data.

Listing 30: Reviews.vue

PageNotFound.vue :

Nesta view é disponibilizada a representação de uma rota inexistente no sistema,i.e, 404 Page Not Found .

Listing 31: PageNotFound.vue

7.3 Results



Figura 5: Home Page Header

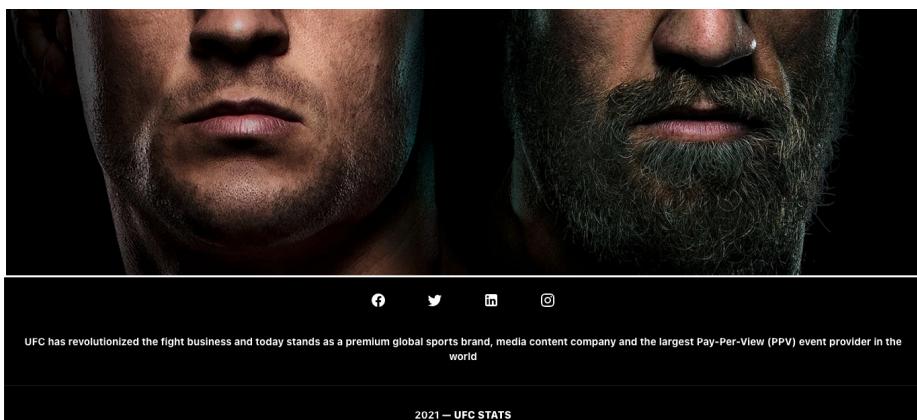


Figura 6: Home Page Footer



Figura 7: Navigation Bar

UFC EVENTS			
NAME	LOCATION	DATE	Fights
UFC 4 Revenge of the Warriors	USA,Tulsa,Oklahoma	December 16 1994	8
UFC 2 No Way Out	USA,Denver,Colorado	March 11 1994	15
UFC 3 The American Dream	USA,Charlotte,North Carolina	September 09 1994	6
UFC Ultimate Ultimate 95	USA,Denver,Colorado	December 16 1995	7
UFC 5 The Return of the Beast	USA,Charlotte,North Carolina	April 07 1995	10
UFC 6 Clash of the Titans	USA,Casper,Wyoming	July 14 1995	8
UFC 7 The Brawl in Buffalo	USA,Buffalo,New York	September 08 1995	9
UFC Ultimate Ultimate 96	USA,Birmingham,Alabama	December 07 1996	7
UFC 8 David vs Goliath	Puerto Rico,San Juan	February 16 1996	8

Figura 8: Events



UFC 4 REVENGE OF THE WARRIORS

LOCATION: USA,Tulsa,Oklahoma DATE: December 16 1994

Result	Fighter	KD	STR	TD	SUB	Method	Round	Time	Bout	Referee
W	Dan Severn	0	0 of 0	2 of 5	1	Submission	1	1:45	Open Weight Bout	John McCarthy
L	Anthony Macias	0	8 of 9	0 of 0	1					
W	Royce Gracie	0	0 of 4	0 of 0	3	Submission	1	15:49	UFC 4 Tournament Title Bout	John McCarthy
L	Dan Severn	0	4 of 7	1 of 1	3					
W	Dan Severn	0	0 of 0	1 of 1	1	Submission	1	0:52	Open Weight Bout	John McCarthy
L	Marcus Bossett	0	1 of 2	0 of 0	1					

Figura 9: Event

UFC FIGHTERS										
NAME		NICKNAME	HEIGHT	WEIGHT	REACH	STANCE	W	L	D	BELT ↓
Alexander Volkanovski	The Great	5' 6"	145 lbs	71.0	Orthodox	22	1	0	Gold	
Alijain Sterling	Funk Master	5' 7"	135 lbs	71.0	Orthodox	20	3	0	Gold	
Amanda Nunes	Lioness	5' 8"	145 lbs	69.0	Orthodox	21	4	0	Gold	
Brandon Moreno	The Assassin Baby	5' 7"	125 lbs	70.0	Orthodox	19	5	2	Gold	
Charles Oliveira	Do Bronx	5' 10"	155 lbs	74.0	Orthodox	31	8	0	Gold	
Francis Ngannou	The Predator	6' 4"	250 lbs	83.0	Orthodox	16	3	0	Gold	
Israel Adesanya	The Last Stylebender	6' 4"	185 lbs	80.0	Switch	21	1	0	Gold	
Jan Blachowicz	--	6' 2"	205 lbs	78.0	Orthodox	28	8	0	Gold	
Kamaru Usman	The Nigerian Nightmare	6' 0"	170 lbs	76.0	Switch	19	1	0	Gold	

Figura 10: Fighters

UFC FIGHTERS										
NAME		NICKNAME	HEIGHT	WEIGHT	REACH	STANCE	W	L	D	BELT ↓
Alexander Volkanovski	The Great	5' 6"	1.71 m	145 lbs	71.0	Orthodox	22	1	0	Gold

Figura 11: Tooltip Weight

NAME	NICKNAME	HEIGHT	WEIGHT	REACH	STANCE	W	L	D	BELT
Alexander Volkanovski	The Great	5' 6	145 lbs	65.77 Kg	Orthodox	22	1	0	

Figura 12: Tooltip Height

NAME	NICKNAME	HEIGHT	WEIGHT	REACH	STANCE	W	L	D	BELT	
Alexander Volkanovski	The Great	5' 6	145 lbs	71.0	180.34 cm	Orthodox	22	1	0	

Figura 13: Tooltip Reach

ANDERSON SILVA
"THE SPIDER"
RECORD: 34-11-0

Fighter Details	Career Statistics	Help Guide
Height: 6' 2 Weight: 185 lbs Reach: 77.0 Stance: Southpaw DOB: Apr 14 1975	SLPm: 3.05 Str. Acc: 61.0% SAPM: 2.05 Str. Def: 60.0% TD Avg: 0.5 TD Acc: 77.0% TD Def: 69.0% Sub. Avg: 0.8	SLPm - Significant Strikes Landed per Minute Str. Acc. - Significant Striking Accuracy SAPM - Significant Strikes Absorbed per Minute Str. Def. - Significant Strike Defence TD Avg. - Average Takedowns Landed per 15 minutes TD Acc. - Takedown Accuracy TD Def. - Takedown Defense

Figura 14: Anderson Silva - Fighter Details

Result	Fighter	KD	STR	TD	SUB	Event	Method	Round	Time	Bout	Referee
W	Uriah Hall	2	60 of 111	0 of 0	0	UFC Fight Night Hall vs Silva	KO/TKO	4	1:24	Middleweight Bout	Herb Dean
L	Anderson Silva	0	53 of 110	0 of 0	0	October 31 2020					
W	Jared Cannonier	1	17 of 31	0 of 0	0	UFC 237 Namajunas vs Andrade	KO/TKO	1	4:47	Middleweight Bout	Herb Dean
L	Anderson Silva	0	8 of 15	0 of 0	0	May 11 2019					

Figura 15: Anderson Silva - Fighter Details 2



Figura 16: Predictions

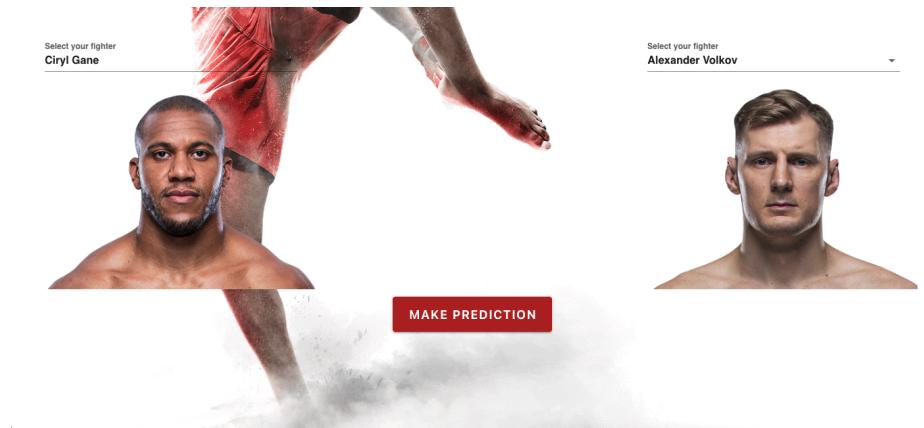


Figura 17: Predictions 2

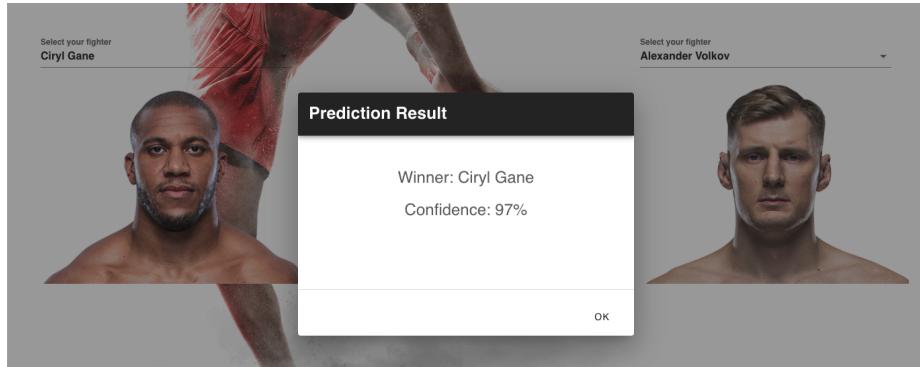


Figura 18: Prediction Result

PREDICTION HISTORY				
Favourite	Underdog	Winner	Confidence	Date
Ciryl Gane	Alexander Volkov	Ciryl Gane	97 %	June 28, 2021 at 04:26
Charles Oliveira	Paul Felder	Paul Felder	96 %	June 28, 2021 at 03:44
Tanner Boser	Ovince Saint Preux	Ovince Saint Preux	99 %	June 27, 2021 at 21:11
Kennedy Nzechukwu	Daniel McWilliams	Kennedy Nzechukwu	100 %	June 27, 2021 at 21:11
Tim Means	Nicolas Dalby	Nicolas Dalby	81 %	June 27, 2021 at 21:10
Andre Fili	Daniel Pineda	Andre Fili	94 %	June 27, 2021 at 21:10
Cody Garbrandt	TJ Dillashaw	Cody Garbrandt	93 %	June 27, 2021 at 21:05
Khabib Nurmagomedov	Kamaru Usman	Kamaru Usman	93 %	June 27, 2021 at 21:05
AJ Iaquinta	Conor McGregor	Conor McGregor	92 %	June 27, 2021 at 21:03

Figura 19: Prediction History



RECORD BOOK

Career	Fight	Event	Location	Referee	Title Holders																																				
UFC TOTAL FIGHTS			MMA TOTAL FIGHTS																																						
<table border="1"><thead><tr><th>RNK</th><th>FIGHTER</th><th>TOTAL</th></tr></thead><tbody><tr><td>1</td><td>Donald Cerrone</td><td>37</td></tr><tr><td>2</td><td>Jim Miller</td><td>37</td></tr><tr><td>3</td><td>Andrei Arlovski</td><td>35</td></tr><tr><td>4</td><td>Demian Maia</td><td>33</td></tr><tr><td>5</td><td>Jeremy Stephens</td><td>33</td></tr></tbody></table>			RNK	FIGHTER	TOTAL	1	Donald Cerrone	37	2	Jim Miller	37	3	Andrei Arlovski	35	4	Demian Maia	33	5	Jeremy Stephens	33	<table border="1"><thead><tr><th>RNK</th><th>FIGHTER</th><th>TOTAL</th></tr></thead><tbody><tr><td>1</td><td>Travis Fulton</td><td>316</td></tr><tr><td>2</td><td>Shannon Ritch</td><td>139</td></tr><tr><td>3</td><td>Dan Severn</td><td>121</td></tr><tr><td>4</td><td>Jeremy Horn</td><td>118</td></tr><tr><td>5</td><td>Ikuhisa Minowa</td><td>110</td></tr></tbody></table>			RNK	FIGHTER	TOTAL	1	Travis Fulton	316	2	Shannon Ritch	139	3	Dan Severn	121	4	Jeremy Horn	118	5	Ikuhisa Minowa	110
RNK	FIGHTER	TOTAL																																							
1	Donald Cerrone	37																																							
2	Jim Miller	37																																							
3	Andrei Arlovski	35																																							
4	Demian Maia	33																																							
5	Jeremy Stephens	33																																							
RNK	FIGHTER	TOTAL																																							
1	Travis Fulton	316																																							
2	Shannon Ritch	139																																							
3	Dan Severn	121																																							
4	Jeremy Horn	118																																							
5	Ikuhisa Minowa	110																																							
Figura 20: Career Stats																																									



RECORD BOOK

Career	Fight	Event	Location	Referee	Title Holders																								
FASTEST FINISH			FASTEST KO/TKO																										
<table border="1"><thead><tr><th>RNK</th><th>FIGHTER</th><th>TOTAL</th></tr></thead><tbody><tr><td>1</td><td>Jorge Masvidal VS Ben Askren</td><td>0:05</td></tr><tr><td colspan="3">July 06 2019 UFC 239 Jones vs Santos</td></tr><tr><td>2</td><td>Duane Ludwig VS Jonathan Goulet</td><td>0:06</td></tr></tbody></table>			RNK	FIGHTER	TOTAL	1	Jorge Masvidal VS Ben Askren	0:05	July 06 2019 UFC 239 Jones vs Santos			2	Duane Ludwig VS Jonathan Goulet	0:06	<table border="1"><thead><tr><th>RNK</th><th>FIGHTER</th><th>TOTAL</th></tr></thead><tbody><tr><td>1</td><td>Jorge Masvidal VS Ben Askren</td><td>0:05</td></tr><tr><td colspan="3">July 06 2019 UFC 239 Jones vs Santos</td></tr><tr><td>2</td><td>Duane Ludwig VS Jonathan Goulet</td><td>0:06</td></tr></tbody></table>			RNK	FIGHTER	TOTAL	1	Jorge Masvidal VS Ben Askren	0:05	July 06 2019 UFC 239 Jones vs Santos			2	Duane Ludwig VS Jonathan Goulet	0:06
RNK	FIGHTER	TOTAL																											
1	Jorge Masvidal VS Ben Askren	0:05																											
July 06 2019 UFC 239 Jones vs Santos																													
2	Duane Ludwig VS Jonathan Goulet	0:06																											
RNK	FIGHTER	TOTAL																											
1	Jorge Masvidal VS Ben Askren	0:05																											
July 06 2019 UFC 239 Jones vs Santos																													
2	Duane Ludwig VS Jonathan Goulet	0:06																											
Figura 21: Fight Stats																													



RECORD BOOK

Career	Fight	Event	Location	Referee	Title Holders
KO/TKO WINS					
SUBMISSION WINS					
RNK	EVENT	TOTAL	RNK	EVENT	TOTAL
1	UFC 92 The Ultimate 2008 December 27 2008	8	1	UFC 2 No Way Out March 11 1994	10
2	UFC on FOX Henderson vs Melendez April 20 2013	8	2	UFC on FUEL TV Nogueira vs Werdum June 08 2013	8
3	UFC 199 Rockhold vs Bisping 2	8	3	UFC 4 Revenge of the Warriors	7

Figura 22: Event Stats



RECORD BOOK

Career	Fight	Event	Location	Referee	Title Holders
TOTAL FIGHTS					
LOCATION					
RNK	LOCATION	TOTAL	RNK	LOCATION	TOTAL
1	USA,Las Vegas,Nevada	159	1	USA,Atlantic City,New Jersey	9
2	United Arab Emirates,Abu Dhabi,Abu Dhabi	13	2	United Kingdom,London,England	11
3			3		
4			4		

Figura 23: Location Stats



RECORD BOOK

Career	Fight	Event	Location	Referee	Title Holders
TOTAL FIGHTS					
REFEREES					
RNK	REFEREE	TOTAL			
1	Herb Dean	885			
2	John McCarthy	639			
3	Dan Miragliotta	405			
4	Mario Yamasaki	392			
5	Marc Goddard	351			

Figura 24: Referee Stats



RECORD BOOK

Career	Fight	Event	Location	Referee	Title Holders
CHAMPIONS					
WEIGHT DIVISION					
WEIGHT	FIGHTER	RECORD			
115 lbs	Rose Namajunas	11-4-0			
125 lbs	Valentina Shevchenko	21-3-0			
125 lbs	Brandon Moreno	19-5-2			
135 lbs	AjjaMain Sterling	20-3-0			
145 lbs	Alexander Volkanovski	22-1-0			

Figura 25: Title Holders

RATE OUR SITE

If you enjoy using *UFC STATS*, please take a few seconds to rate your experience with the website. It really helps!

Name _____ 0 / 15

E-mail _____

Review _____ 0 / 150

VALIDATE **RESET FORM** **RESET VALIDATION**

Figura 26: Rate Us

RATE OUR SITE

If you enjoy using *UFC STATS*, please take a few seconds to rate your experience with the website. It really helps!

Name _____ 0 / 15
Name is required

E-mail _____
E-mail is required

Review _____ 0 / 150
Review is required

VALIDATE **RESET FORM** **RESET VALIDATION**

Figura 27: Rate Error

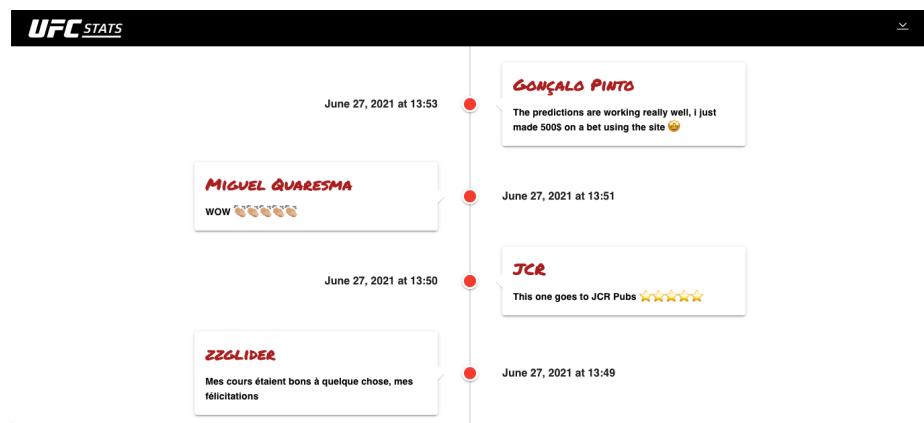


Figura 28: Reviews

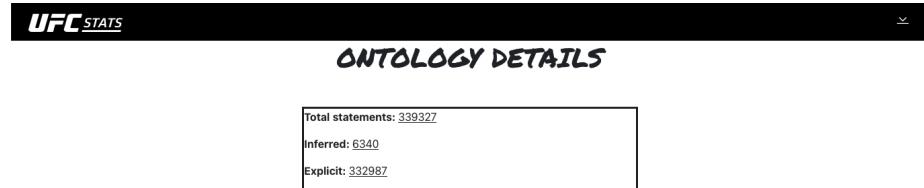


Figura 29: Ontology Details

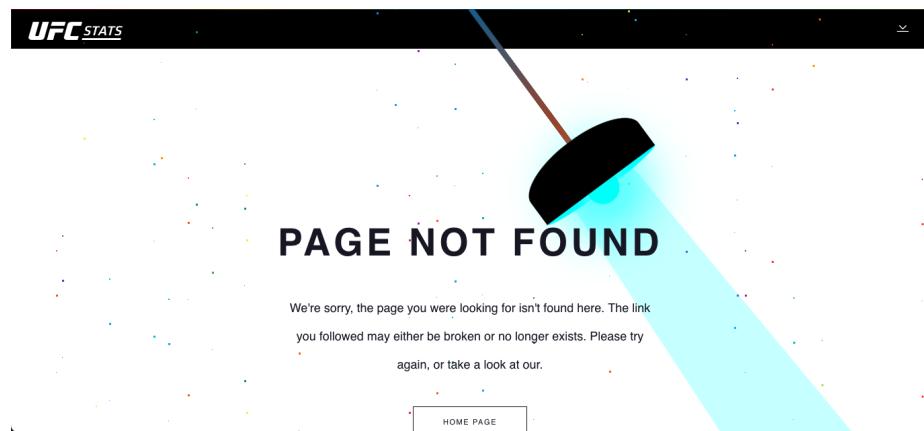


Figura 30: Page Not Found

8 How to start

De modo a facilitar a execução completa do projecto, criou-se um pequeno script *ufc-script.scpt* compilado criado com o editor de scripts da Apple. Ele é escrito em AppleScript, uma linguagem de script de automação usada por computadores Mac, sendo que os mesmos podem ser gravados manualmente ou gerados por acções de gravação .

Para tirar partido desse script desenvolvido , existe a necessidade de resolver algumas dependências :

- iTerm - Terminal utilizado pelos criadores do sistema .
- Node.js
- Nodemon
- GraphDB

O comando de execução é o seguinte : **osascript ufc-script.scpt**

```
tell application "iTerm"
    --Create initial window
    create window with default profile

    --Create fours tab in the initial window
    tell current window
        create tab with default profile
        create tab with default profile
        create tab with default profile
    end tell

    --API SERVER
    tell current session of tab 1 of current window
        write text "cd
            /Users/etiennecosta/Desktop/Mestrado/PLC/PRC/UFC-PREDICTIONS/ufc-api"
        write text "nodemon npm run start"
    end tell

    --MongoDB SERVER
    tell current session of tab 2 of current window
        write text "cd
            /Users/etiennecosta/Desktop/Mestrado/PLC/PRC/UFC-PREDICTIONS/ufc-mongodb"
        write text "nodemon npm run serve"
    end tell

    --APP SERVER
    tell current session of tab 3 of current window
        write text "cd
            /Users/etiennecosta/Desktop/Mestrado/PLC/PRC/UFC-PREDICTIONS/ufc-app"
```

```

        write text "npm run serve"
end tell

--RUN GraphDB
tell current session of tab 4 of current window
    write text "open /Applications/'GraphDB Free.app' "
end tell

end tell

```

Listing 32: ufc-script.scpt

9 Deployment

De modo a tornar a aplicação acessível à comunidade interessada, decidiu-se recorrer ao *Docker*. O Docker é um conjunto de produtos de plataforma como serviço que usam virtualização de nível de sistema operativos para entregar software em pacotes chamados containers. Os containers são isolados uns dos outros e agrupam os seus próprios softwares, bibliotecas e arquivos de configuração.

De seguida são relembrados os serviços implementados, tendo para cada um deles um *Dockerfile* associado.

9.1 GraphDB

A criação e execução deste *Dockerfile* permite-nos ter uma imagem do *GraphDB* com a nossa ontologia sobre o ufc.

```

FROM openjdk:8-jdk

ARG version=free-9.8.0
ARG dataFile=ufc-inf.ttl

ENV GRAPHDB_PARENT_DIR=/opt/graphdb
ENV GRAPHDB_HOME=${GRAPHDB_PARENT_DIR}/home

ENV GRAPHDB_INSTALL_DIR=${GRAPHDB_PARENT_DIR}/dist

ENV UFC=${GRAPHDB_HOME}/data/repositories/UFC

ADD ./graphdb-${version}-dist.zip /tmp

RUN mkdir -p ${GRAPHDB_PARENT_DIR} && \
    cd ${GRAPHDB_PARENT_DIR} && \
    unzip /tmp/graphdb-${version}-dist.zip && \
    rm /tmp/graphdb-${version}-dist.zip && \
    mv graphdb-${version} dist && \

```

```

mkdir -p ${GRAPHDB_HOME} && \
mkdir -p ${UFC}

COPY ./UFC-config.ttl ${UFC}/config.ttl
COPY ./${dataFile} /tmp

RUN sed -i 's|# graphdb.home.data =|graphdb.home.data = ../home/data|' \
${GRAPHDB_INSTALL_DIR}/conf/graphdb.properties

RUN /opt/graphdb/dist/bin/loadrdf -f -c ${UFC}/config.ttl -m parallel \
/tmp/${dataFile}

ENV PATH=${GRAPHDB_INSTALL_DIR}/bin:$PATH

CMD ["/opt/graphdb/dist/bin/graphdb",
"-Dgraphdb.home=/opt/graphdb/home", "-s"]

EXPOSE 7200

```

Listing 33: GraphDB Dockerfile

```

#
# RDF4J configuration template for a GraphDB Free repository
#
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rep: <http://www.openrdf.org/config/repository#>.
@prefix sr: <http://www.openrdf.org/config/repository/sail#>.
@prefix sail: <http://www.openrdf.org/config/sail#>.
@prefix owllim: <http://www.ontotext.com/trree/owlim#>.

[] a rep:Repository ;
    rep:repositoryID "UFC" ;
    rdfs:label "UFC" ;
    rep:repositoryImpl [
        rep:repositoryType "graphdb:FreeSailRepository" ;
        sr:sailImpl [
            sail:sailType "graphdb:FreeSail" ;

            owllim:base-URL "http://example.org/owlim#" ;
            owllim:defaultNS "" ;
            owllim:entity-index-size "10000000" ;
            owllim:entity-id-size "32" ;
            owllim:imports "" ;
            owllim:repository-type "file-repository" ;
            owllim:ruleset "rdfsplus-optimized" ;
            owllim:storage-folder "storage" ;

            owllim:enable-context-index "false" ;

```

```

owlim:enablePredicateList "true" ;

owlim:in-memory-literal-properties "true" ;
owlim:enable-literal-index "true" ;

owlim:check-for-inconsistencies "false" ;
owlim:disable-sameAs "true" ;
owlim:query-timeout "0" ;
owlim:query-limit-results "0" ;
owlim:throw-QueryEvaluationException-on-timeout "false" ;
owlim:read-only "false" ;
]
].

```

Listing 34: Ficheiro de configuração config.ttl

É de salientar que na directória deve estar presente o *turtle* que se pretende inserir, bem como uma distribuição do graphdb .

9.2 UFC API

Dockerfile associado à API que tem como objectivo fazer as queries ao nosso servidor que contém a imagem do *GraphDB*.

```

FROM node:latest
WORKDIR /ufc-api
COPY . /ufc-api
RUN npm install
CMD ["npm", "start"]

```

Listing 35: UFC-API Dockerfile

9.3 UFC Mongo

Dockerfile associado à API que tem como objectivo fazer garantir a persistência de dados que não estão diretamente ligados a nossa ontologia .

```
FROM node:latest
WORKDIR /ufc-mongo
COPY . /ufc-mongo
RUN npm install
CMD ["npm", "start"]
EXPOSE 8078
```

Listing 36: UFC-Mongo Dockerfile

9.4 UFC APP

Dockerfile associado à aplicação desenvolvida.

```
FROM node:latest
COPY . /app
WORKDIR /app
RUN npm install && npm run build

FROM nginx
RUN mkdir /app
COPY --from=0 /app/dist /app
COPY nginx.conf /etc/nginx/nginx.conf
CMD ["nginx", "-g", "daemon off;"]
```

Listing 37: UFC-APP Dockerfile

```
user nginx;
worker_processes 1;
error_log /var/log/nginx/error.log warn;
pid      /var/run/nginx.pid;
events {
    worker_connections 1024;
}
http {
    include      /etc/nginx/mime.types;
    default_type application/octet-stream;
    log_format main '$remote_addr - $remote_user [$time_local] "$request" ,
                      '$status $body_bytes_sent "$http_referer" ,
                      '"$http_user_agent" "$http_x_forwarded_for"';
    access_log /var/log/nginx/access.log main;
    sendfile      on;
    keepalive_timeout 65;
    server {
        listen      7777;
        server_name localhost;
        location / {
            add_header 'Access-Control-Allow-Origin' '*';
            add_header 'Access-Control-Allow-Methods' 'GET, POST, PUT, DELETE,
OPTIONS';
            add_header 'Access-Control-Allow-Headers'
                        'Accept,Authorization,Cache-Control,Content-Type,DNT,
If-Modified-Since,Keep-Alive,
Origin,User-Agent,X-Requested-With,Content-Length';
            if ($request_method = 'OPTIONS') {
                add_header 'Access-Control-Max-Age' 1728000;
                add_header 'Content-Type' 'text/plain; charset=utf-8';
                add_header 'Content-Length' 0;
                return 204;
            }
            root   /app;
            index index.html;
            try_files $uri $uri/ /index.html;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root   /usr/share/nginx/html;
        }
    }
}
```

Listing 38: NGINX.conf File

9.5 Docker Compose

Compose é uma ferramenta para definir e executar aplicativos Docker de vários containers. Com o Compose, usa-se um arquivo YAML para configurar os serviços das aplicações.

Portanto aos interessados em correr a aplicação, basta terem instaladas as dependências *Docker* e *Docker-Compose* e executar :

```
docker-compose -f docker-compose.yml up
```

Listing 39: docker-compose command

```
version: '3'
services:
  mongo:
    image: mongo
    volumes:
      - db-data:/data/db
    ports:
      - "27017:27017"
  graphdb-ufc:
    image: etennecosta/graphdb-ufc
    ports:
      - "7200:7200"
  sparql-api-ufc:
    image: etennecosta/sparql-api-ufc:latest
    restart: on-failure
    links:
      - graphdb-ufc
    ports:
      - "8079:8079"
  mongodb-ufc:
    image: etennecosta/mongodb-ufc
    restart: on-failure
    links:
      - mongo
    ports:
      - "8078:8078"
  interface-vue-ufc:
    image: etennecosta/interface-vue-ufc
    links:
      - mongodb-ufc
    ports:
      - "8080:7777"
volumes:
  db-data:
```

Listing 40: docker-compose.yml

10 Conclusion

O processo de realização deste projecto foi dividido em diversas etapas, tendo cada uma a sua complexidade associada, mas apesar das dificuldades encontradas durante a realização conseguiu-se implementar na totalidade todas as funcionalidades desejadas. Todo software está passível de melhorias, portanto, como trabalho futuro seria desejável ter a capacidade de executar o script de criação e integração semanalmente de modo a garantir que a informação esteja sempre actualizada, e também será desenvolvido o próprio algoritmo de previsão de modo a que os utilizadores registem uma versão sua de lutador e tentar prevar os resultados com os diversos lutadores da companhia .

Referências

- [1] MongoDB Homepage, <http://www.MongoDB.com>.
- [2] Nodejs Homepage, <https://nodejs.org/en/>.
- [3] Vue.js Homepage, <https://vuejs.org/>.
- [4] Vuetify Homepage, <https://vuetifyjs.com/en/>.
- [5] UFC, <https://www.ufc.com/>.
- [6] Docker, <https://www.docker.com/>.
- [7] SPARQL Query Language for RDF, <https://www.w3.org/TR/rdf-sparql-query/>